

IMPLEMENTASI SISTEM KLASIFIKASI SAMPAH ORGANIK DAN ANORGANIK DENGAN METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Fungki Pandu Fantara
NIM: 145150301111009



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI SISTEM KLASIFIKASI SAMPAH ORGANIK DAN ANORGANIK
DENGAN METODE JARINGAN SARAF TIRUAN *BACKPROPAGATION*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh:

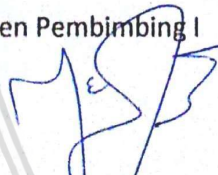
Fungki Pandu Fantara

NIM: 145150301111009

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Juli 2018

Telah diperiksa dan disetujui oleh:

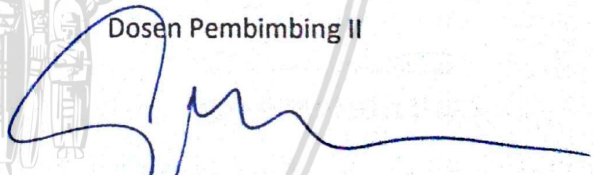
Dosen Pembimbing I



Dahnial Syauqy, S.T., M.T., M.Sc.

NIK: 2016078704231002

Dosen Pembimbing II



Gembong Edhi Setyawan, S.T., M.T.

NIK: 2012087612011001

Mengetahui



Setiajiwan, S.T., M.T., Ph.D.

Tri Astoto Kurmiawan, S.T., M.T., Ph.D.

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Juli 2018



Funghi Pandu Fantara

NIM: 145150301111035

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena Rahmat-Nya lah penulis dapat menyelesaikan skripsi ini. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam menempuh ujian Sarjana Fakultas Ilmu Komputer. Judul skripsi yang disusun adalah: **"IMPLEMENTASI SISTEM KLASIFIKASI SAMPAH ORGANIK DAN ANORGANIK DENGAN METODE JARINGAN SARAF TIRUAN BACKPROPAGATION"**.

Dalam menyusun skripsi ini ada banyak kesulitan dan hambatan yang dialami oleh peneliti, tetapi semua itu telah dapat diatasi dengan baik berkat dukungan dan bantuan dari berbagai pihak. Untuk itulah pada kesempatan ini peneliti mengucapkan terima kasih kepada:

1. Bapak dan Ibu serta keluarga besar peneliti yang telah memberikan dukungan, semangat serta doanya hingga terselesaikannya skripsi ini.
2. Bapak Dahnia Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingannya kepada peneliti selama proses pengerjaan skripsi sehingga dapat terselesaikan dengan baik.
3. Bapak Gembong Edhi Setyawan, S.T, M.T selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingannya selama proses pembuatan laporan skripsi sehingga skripsi ini dapat terselesaikan dengan baik.
4. Teman-teman dari komunitas The-Ex yang senantiasa memberikan semangat dan motivasi selama proses pengerjaan skripsi ini.
5. Seluruh pimpinan dekanat beserta jajarannya, seluruh civitas akademik, serta seluruh teman-teman dari keminatan Teknik Komputer angkatan 2014 Fakultas Ilmu Komputer Universitas Brawijaya yang telah banyak membantu peneliti dalam proses pengerjaan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, 25 Juli 2018

Penulis

fungkipandu@yahoo.com

ABSTRAK

Fungki Pandu Fantara, Implementasi Sistem Klasifikasi Sampah Organik Dan Anorganik Dengan Metode Jaringan Saraf Tiruan *Backpropagation*

Pembimbing: Dahnia Syauqy, S.T., M.T., M.Sc. dan Gembong Edhi Setyawan, S.T., M.T.

Sampah organik dan anorganik memiliki lama penguraian yang berbeda. Sampah organik memiliki waktu penguraian yang lebih cepat dibandingkan dengan sampah anorganik. Oleh sebab itu sampah organik dan anorganik memiliki cara penanganan daur ulang yang berbeda pula. Pemilahan sampah sebelum ditampung ke Tempat Pembuangan Akhir (TPA) sangat penting untuk mengurangi jumlah penimbunan sampah yang terus meningkat setiap tahunnya. Penelitian ini meneliti tentang implementasi sistem klasifikasi sampah organik dan anorganik dengan menggunakan metode *backpropagation*. Arsitektur jaringan saraf tiruan yang diterapkan yaitu 3 unit neuron pada lapis masukan, 1 lapis tersembunyi yang mempunyai 4 unit neuron, serta 1 unit pada lapis keluaran. Pelatihan data tidak dilakukan pada sistem yang dibangun namun pada sistem tambahan untuk mencari bobot, sehingga sistem yang dibangun hanya melakukan prediksi data secara langsung dari pembacaan sensor. Berdasarkan penelitian ini, sistem dapat dibangun menggunakan 3 buah sensor yang digunakan sebagai data masukan yaitu sensor *Light Dependent Resistor* (LDR), *proximity* induktif, dan *proximity* kapasitif dan sebuah keluaran berupa servo yang dapat membuka tutup tempat sampah secara otomatis berdasarkan hasil klasifikasi yang dilakukan oleh sistem. Sistem memiliki akurasi 90% dengan kinerja setiap melakukan prediksi membutuhkan waktu rata-rata 42,9ms.

Kata kunci: klasifikasi, jaringan saraf tiruan, *backpropagation*, sampah

ABSTRACT

Fungki Pandu Fantara, *Implementation of Organic and Inorganic Classification System Using Backpropagation Neural Network*

Supervisors: Dahnial Syauqy, S.T., M.T., M.Sc. and Gembong Edhi Setyawan, S.T., M.T.

Organic and inorganic waste have different decomposition time. Organic waste has longer decomposition time than inorganic waste. So, organic and inorganic waste have a different ways of handling recycling. Sorting garbage before being accommodated to landfills (TPA) is very important to reduce the amount of garbage dump that keeps increasing every year. This research examines the implementation of classification system of organic and inorganic waste by using artificial neural network method backpropagation. Artificial neural network architecture applied is 3 neurons in the input layer, 1 layer hidden with 4 neurons, and 1 neurons on the output layer. Data training is not performed on built systems but on additional systems to search for weights, so the built system only predicts data directly from sensor readings. Based on this research, the system can be built using 3 sensors which are used as input data, they are: Light Dependent Resistor (LDR), inductive proximity, and capacitive proximity and a servo output which can open the lid automatically based on the classification result done by system. The system has 90% accuracy with the performance of each prediction takes 42.9ms average time.

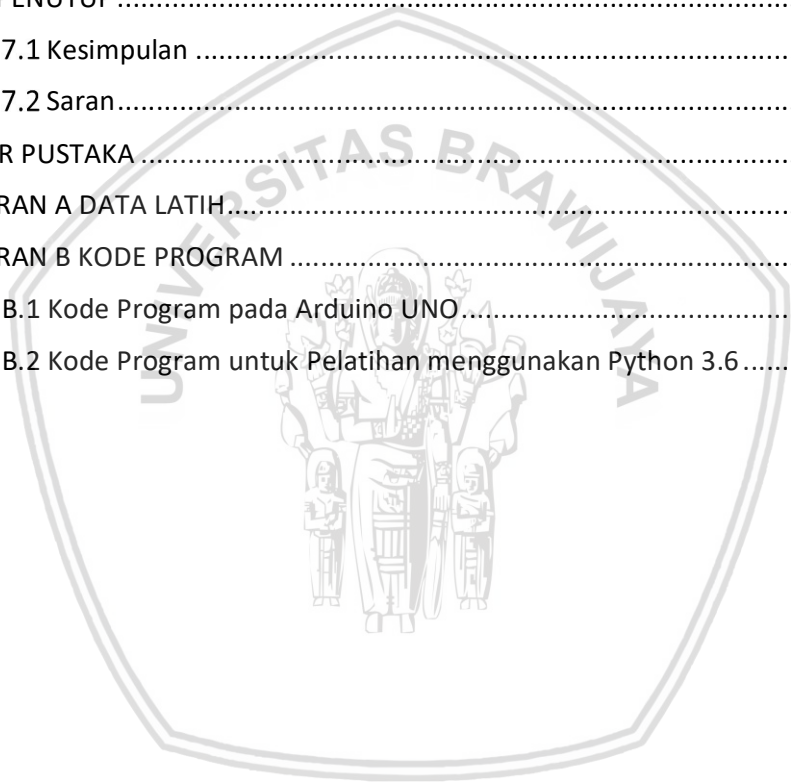
Keywords: *classification, neural network, backpropagation, waste*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori.....	7
2.2.1 Sampah.....	7
2.2.2 Klasifikasi Sampah.....	7
2.2.3 Pengenalan Pola	8
2.2.4 Jaringan Saraf Tiruan.....	8
2.2.5 <i>Backpropagation</i>	8
2.2.6 Pelatihan Backpropagation	9
2.2.7 Pengukuran Kesalahan dari Perubahan Bobot.....	10
2.2.8 Pengukuran Kinerja Klasifikasi.....	11
2.2.9 Rata-rata (<i>average</i>)	11
2.2.10 Rangkaian Pembagi Tegangan	12
2.2.11 ADC (<i>Analog to Digital Converter</i>)	12
2.2.12 <i>Light Dependent Resistor</i> (LDR)	13

2.2.13 Sensor <i>Proximity</i> Induktif	13
2.2.14 Sensor <i>Proximity</i> Kapasitif	14
2.2.15 Pulse Width Modulation (PWM).....	14
BAB 3 METODOLOGI PENELITIAN.....	16
3.1 Studi Pustaka	17
3.2 Rekayasa Kebutuhan	17
3.3 Perancangan Sistem	18
3.4 Implementasi Sistem	18
3.5 Pengujian dan Analisis.....	19
3.6 Penarikan Kesimpulan.....	19
BAB 4 REKAYASA KEBUTUHAN	20
4.1 Gambaran Umum Sistem	20
4.1.1 Perspektif Sistem	20
4.1.2 Ruang Lingkup.....	20
4.1.3 Karakteristik Pengguna	20
4.2 Analisis Kebutuhan Sistem	20
4.2.1 Kebutuhan Fungsional.....	21
4.2.2 Analisis Kebutuhan Non-fungsional	21
4.3 Batasan Sistem	26
BAB 5 PERANCANGAN dan IMPLEMENTASI.....	27
5.1 Perancangan Sistem	27
5.1.1 Perancangan Perangkat keras	27
5.1.2 Perancangan Perangkat Lunak	36
5.1.3 Perancangan Jaringan Saraf Tiruan.....	39
5.2 Implementasi Prototipe Alat	41
5.2.1 Implementasi Perangkat Keras.....	41
5.2.2 Implementasi Perangkat Lunak	42
BAB 6 PENGUJIAN	48
6.1 Pengujian Fungsional Setiap Sensor	48
6.1.1 Sensor LDR (Light Dependent Resistor)	48
6.1.2 Sensor <i>Proximity</i> Induktif	49
6.1.3 Sensor <i>Proximity</i> Kapasitif	51

6.2 Pengujian Kinerja Sistem	52
6.2.1 Tujuan Pengujian	52
6.2.2 Prosedur Pengujian	52
6.2.3 Hasil Pengujian	52
6.3 Pengujian Performa	54
6.3.1 Durasi Pelatihan Data	54
6.3.2 Analisis	55
6.3.3 Durasi Prediksi	55
BAB 7 PENUTUP	57
7.1 Kesimpulan	57
7.2 Saran	57
DAFTAR PUSTAKA	58
LAMPIRAN A DATA LATIH	60
LAMPIRAN B KODE PROGRAM	63
B.1 Kode Program pada Arduino UNO	63
B.2 Kode Program untuk Pelatihan menggunakan Python 3.6	65



DAFTAR TABEL

Tabel 2.1	Matriks konfusi 2 kelas	11
Tabel 4.1	Spesifikasi Arduino UNO	22
Tabel 4.2	Spesifikasi LDR	23
Tabel 4.3	Spesifikasi <i>Proximity</i> Induktif Lj12a3-4-Z	23
Tabel 4.4	Spesifikasi <i>Proximity</i> Kapasitif CR18-8DP	23
Tabel 4.5	Spesifikasi modul <i>Step Up</i> XL6009E	24
Tabel 4.6	Spesifikasi Servo MG-NFC	24
Tabel 4.7	Spesifikasi Komputer	25
Tabel 5.1	Tabel batas nilai minimum dan maksimum nilai ADC berdasarkan perubahan tegangan dari LDR	30
Tabel 5.2	Konfigurasi pin sensor <i>proximity</i> kapasitif	31
Tabel 5.3	Konfigurasi pin modul <i>step up</i> XL6009E dengan Arduino UNO	33
Tabel 5.4	Konfigurasi pin sensor <i>proximity</i> induktif	33
Tabel 5.5	Konfigurasi <i>servo</i> terhadap Arduino UNO	34
Tabel 5.6	Tingkat kesalahan (<i>error</i>) berdasarkan <i>epoch</i> dan <i>learning rate</i> (<i>lr</i>) dengan 3 neuron pada lapisan tersembunyi	46
Tabel 5.7	Tingkat kesalahan (<i>error</i>) berdasarkan <i>epoch</i> dan <i>learning rate</i> (<i>lr</i>) dengan 4 neuron pada lapisan tersembunyi	47
Tabel 5.8	Tingkat kesalahan (<i>error</i>) berdasarkan <i>epoch</i> dan <i>learning rate</i> (<i>lr</i>) dengan 5 neuron pada lapisan tersembunyi	47
Tabel 6.1	Hasil Pembacaan sensor LDR	49
Tabel 6.2	Hasil pembacaan sensor <i>proximity</i> induktif	50
Tabel 6.3	Hasil pembacaan sensor <i>proximity</i> kapasitif	51
Tabel 6.4	Hasil uji kesesuaian sistem	53
Tabel 6.5	Analisis hasil pada matriks konfusi	53
Tabel 6.6	Pengujian durasi pelatihan	54
Tabel 6.7	Hasil pengujian durasi prediksi	56

DAFTAR GAMBAR

Gambar 2.1	Arsitektur jaringan saraf tiruan <i>backpropagation</i>	8
Gambar 2.2	Rangkaian pembagi tegangan	12
Gambar 2.3	Elemen dalam sensor <i>proximity</i> induktif (Menke, 2014).....	14
Gambar 2.4	Sinyal PWM dengan siklus tugas yang bervariasi	15
Gambar 3.1	Diagram alir metode penelitian	16
Gambar 5.1	Diagram blok sistem.....	27
Gambar 5.2	Skema rangkaian sistem	28
Gambar 5.3	Percobaan fungsional sensor <i>proximity</i> induktif dengan tegangan 5V	32
Gambar 5.4	Percobaan fungsional sensor <i>proximity</i> induktif dengan tegangan 6V	33
Gambar 5.5	(a) Desain sistem dan (b) desain tempat sampah	35
Gambar 5.6	Ilustrasi desain sistem	35
Gambar 5.7	(a) Kondisi saat tempat sampah tertutup, (b) kondisi saat sistem buka tutup dari ruang anorganik, (c) kondisi saat sistem membuka tutup dari ruang sampah organik.....	36
Gambar 5.8	Diagram alir sistem	36
Gambar 5.9	Diagram alir pelatihan <i>backpropagation</i>	37
Gambar 5.10	(a) diagram alir fungsi <i>scale_dataset</i> , dan (b) diagram alir fungsi Network	38
Gambar 5.11	Diagram alir fungsi Training.....	39
Gambar 5.12	Rancangan jaringan saraf tiruan	40
Gambar 5.13	Implementasi desain sistem	41
Gambar 5.14	Implementasi sistem	42
Gambar 5.15	Potongan program Arduino.....	43
Gambar 5.16	Potongan program prediksi	43
Gambar 5.17	Potongan program pada fungsi.utama	44
Gambar 5.18	Potongan program fungsi <i>scale_dataset</i> dan fungsi <i>load_dataset</i>	45
Gambar 5.19	Potongan program fungsi pelatihan	46
Gambar 5.20	Arsitektur jaringan saraf tiruan yang digunakan	47
Gambar 6.1	Diagram analisis pengujian durasi selama pelatihan data	55

BAB 1 PENDAHULUAN

1.1 Latar belakang

Sisa-sisa dari proses produksi baik industri ataupun rumah tangga yang sudah tidak terpakai biasa kita sebut dengan sampah, dimana pada umumnya di Indonesia dikelompokkan menjadi 2 jenis, yaitu sampah anorganik dan sampah organik. Sampah yang berasal dari makhluk hidup merupakan sampah organik. Sampah organik mudah membusuk atau terurai hanya dalam waktu kurang dari 6 bulan. Sampah yang berasal dari sumber daya yang tidak dapat diperbaharui merupakan sampah anorganik. Sampah ini memiliki ciri sulit terurai sehingga butuh waktu hingga puluhan tahun agar dapat terurai dengan tanah. Di Indonesia jumlah sampah yang dihasilkan dapat mencapai sekitar 11,330 ton/hari. Jika jumlah penduduk Indonesia di desa mencapai 128,5 juta jiwa dan di kota mencapai 126,3 juta jiwa, maka rata-rata setiap orang menghasilkan sampah sekitar 0,05 kg setiap harinya (Badan Pusat Statistik, 2017).

Sampah organik dan anorganik pada umumnya akan diangkut dan ditimbun pada Tempat Pembuangan Akhir (TPA). Jumlah sampah yang ditimbun pada TPA dapat diminimalkan dengan melakukan daur ulang. Sampah yang didaur ulang dapat mengubah barang yang sebelumnya tidak dapat digunakan menjadi memiliki nilai guna kembali. Daur ulang untuk sampah organik dan anorganik memiliki proses yang berbeda-beda, sehingga perlu adanya pemilahan sebelum dilakukan daur ulang. Upaya pemilahan ini cukup penting untuk mempermudah dan mempercepat proses daur ulang. Pada tahun 2020 memungkinkan jika kebutuhan lahan untuk TPA akan terus meningkat hingga 1.610 hektar jika upaya pemilahan serta pengelolaan sampah sebelum masih sangat kecil. Akibat sulitnya pengadaan lahan tersebut, maka sejak tahun 2014 pemerintah Indonesia telah menggagas sebuah komitmen “Indonesia Bersih Sampah 2020” yang masih berjalan hingga kini (Kementerian Lingkungan Hidup dan Kehutanan, 2015).

Pemilahan sampah yang dilakukan pada tumpukan sampah yang cukup luas di TPA akan lebih sulit dilakukan dibandingkan dengan melakukan pemilahan di sebuah tempat sampah. Selama ini tempat sampah yang tersedia hanya berupa tempat sampah dengan pemilahan yang dilakukan oleh masyarakat secara manual. Seperti yang diketahui bahwa saat ini masih banyak dari masyarakat yang tidak mengerti akan perbedaan jenis sampah yang hendak dibuangnya sehingga meskipun telah disediakan tempat sampah yang membedakan sampah organik dan anorganik, masyarakat akan tetap membuang sampah tersebut di tempat dengan jenis yang tidak sesuai. Hal tersebut tentu akan sangat merepotkan dalam upaya pemilahan sampah di tempat sampah dimana merupakan tempat pertama sampah berkumpul. Oleh sebab perlunya sebuah alat yang dapat membantu masyarakat dalam membedakan jenis sampah sebelum memasukkannya ke tempat sampah dengan metode yang klasifikasi yang akurat.

Pada penelitian sebelumnya yang pernah dilakukan oleh Oladapo (2016), yang membahas tentang pemilahan beberapa jenis benda menggunakan sensor *proximity*. Penelitian tersebut berhasil memisahkan beberapa jenis benda seperti kayu, logam, dan plastik ke tempat yang terpisah. Penelitian tersebut masih berfokus pada skala industri. Kemudian penelitian selanjutnya dilakukan oleh Nuraini (2017) yang meneliti mengenai implementasi dan pengujian klasifikasi sampah rumah tangga dengan menggunakan metode *Naive Bayes*. Penelitian tersebut berhasil menunjukkan bahwa sistem yang dibangun dapat membedakan sampah yang berjenis organik dan anorganik dengan akurasi berkisar antara 87,50-95,00%. Namun sistem yang diterapkan masih memerlukan sambungan secara serial dengan sebuah komputer untuk menjalankan proses klasifikasinya. Penelitian lainnya dilakukan oleh Prasetya (2017) yang melakukan penelitian mengenai cara melakukan prediksi menggunakan metode jaringan saraf tiruan *backpropagation*. Penelitian tersebut berhasil melakukan prediksi dengan *error* yang dihasilkan hanya mencapai 10,93%. Penelitian tersebut dilakukan dengan memasukkan data secara manual dan komputasinya dilakukan pada komputer dengan bantuan aplikasi MATLAB. Berdasarkan semua latar belakang yang telah dikemukakan di atas, peneliti akan melakukan penelitian terkait implementasi dan perancangan sistem yang dapat melakukan klasifikasi jenis sampah organik dan anorganik menggunakan metode jaringan saraf tiruan *backpropagation*.

1.2 Rumusan masalah

Rumusan masalah pada penelitian ini berdasarkan latar belakang yang telah diuraikan di atas antara lain:

1. Bagaimana membangun sistem yang dapat membantu pengguna menentukan jenis sampah organik dan anorganik menggunakan metode jaringan saraf tiruan?
2. Bagaimana akurasi sistem dalam melakukan klasifikasi sampah?
3. Bagaimana kinerja sistem dalam melakukan pelatihan data dan prediksi data?

1.3 Tujuan

Berikut merupakan tujuan dari penelitian ini yang telah ditentukan berdasarkan masalah yang telah dirumuskan sebelumnya.

1. Merancang sebuah sistem yang mampu membedakan jenis-jenis sampah organik dan anorganik dengan menggunakan metode jaringan saraf tiruan.
2. Melakukan pengujian pada akurasi prediksi sistem.
3. Melakukan pengujian terhadap lama waktu yang diperlukan saat melakukan pelatihan data dan prediksi data.

1.4 Manfaat

Berikut merupakan manfaat yang dilihat dari fungsi sistem yang akan diteliti serta diimplementasi.

1. Diharapkan dapat menjadi sarana dan acuan untuk membangun sistem pemilah sampah otomatis dengan kinerja yang lebih baik dan akurasi yang lebih akurat sehingga pengelolaan sampah dapat menjadi lebih efektif dan efisien.
2. Diharapkan penelitian ini dapat menjadi bahan referensi untuk penelitian selanjutnya, ataupun hanya sebagai pengetahuan bagi mahasiswa lain yang ingin mengetahui bagaimana membangun sistem pemilah sampah otomatis dengan hasil keakuratan serta kinerjanya yang telah diuji dan dihitung pada penelitian ini.
3. Diharapkan dapat mempermudah masyarakat untuk menentukan dimana sampah tersebut akan dibuang di tempat yang sejenis serta dapat membantu meningkatkan proses daur ulang.

1.5 Batasan masalah

Agar pembahasan dalam penelitian ini dapat dilakukan secara terarah dan mendapatkan hasil sesuai dengan yang diharapkan, maka perlu diterapkan batasan permasalahan yaitu:

1. Sistem hanya dapat mengklasifikasikan benda ke dalam organik atau anorganik.
2. Sistem melakukan klasifikasi dengan metode jaringan saraf tiruan *backpropagation*.

1.6 Sistematika pembahasan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Bab ini membahas tinjauan pustaka dan dasar teori yang mendukung dalam pembuatan sistem ini. Pada bab ini juga terdapat penelitian-penelitian sebelumnya yang menjadi dasar pembuatan sistem ini.

BAB III Metodologi Penelitian

Bab ini membahas tentang langkah-langkah dalam melakukan penelitian ini, antara lain studi pustaka, pengumpulan data, analisa kebutuhan sistem, perancangan sistem, implementasi sistem, pengujian, dan evaluasi sistem.

BAB IV Analisis Kebutuhan

Bab ini menjelaskan secara keseluruhan kebutuhan sistem yang diperlukan meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan fungsional.

BAB V Perancangan dan Implementasi

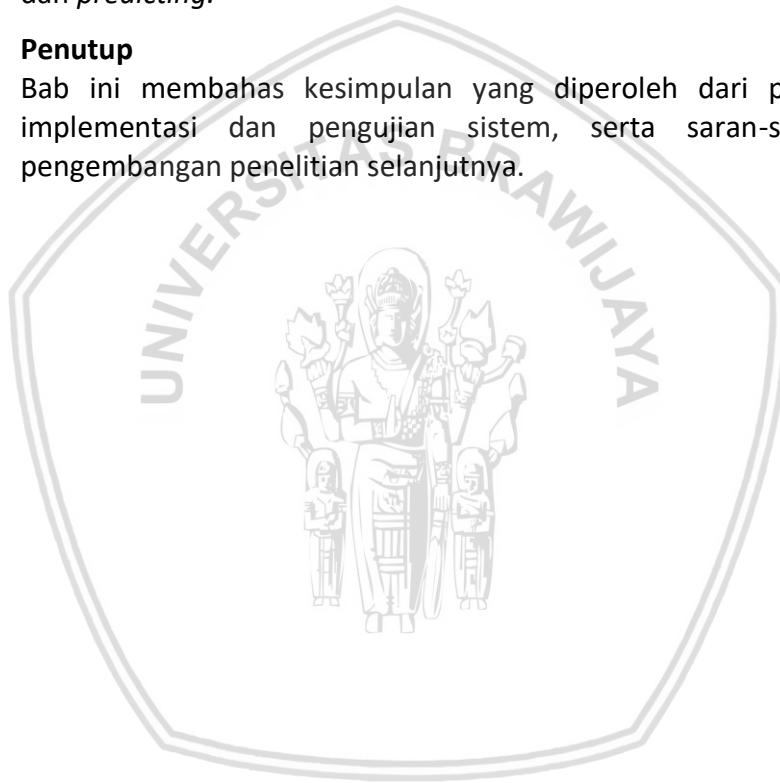
Bab ini membahas tentang proses perancangan sistem perangkat keras dan perangkat lunak, implementasi *prototype*, perangkat keras, dan perangkat lunak.

BAB VI Pengujian

Bab ini membahas tentang cara pengujian, akurasi masing-masing sensor, akurasi sistem, hasil akurasi sistem, serta lama waktu *training* dan *predicting*.

BAB VII Penutup

Bab ini membahas kesimpulan yang diperoleh dari perancangan, implementasi dan pengujian sistem, serta saran-saran untuk pengembangan penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Oladapo, B. I., dkk. (2016) pernah melakukan penelitian untuk memilah otomatis terhadap berbagai macam benda menggunakan sensor *Proximity*. Mesin pemilah otomatis yang dikembangkan tersebut dapat menggabungkan fleksibilitas dan memisahkan benda berdasarkan logam dan non logam, serta mengarahkan objek tersebut secara otomatis ke tempat yang sejenis dengan menggunakan roda berjalan. Mesin tersebut berjalan dengan pengaturan *Programmable Logic Controller* (PLC) dan sensor jarak kapasitif untuk mendeteksi berbagai nilai objek yang dapat menjadi acuan dalam memisahkan jenis benda. Hasil analisa menunjukkan bahwa mesin tersebut dapat membedakan beberapa jenis benda yaitu plastik, kayu dan baja berdasarkan lama dari elektromagnetik ketika ditembakkan dari benda dan dikembalikan ke sensor dengan waktu (secara urut) 9,90 detik, 14,07 detik dan 18,65 detik (Oladapo, 2016). Maka dapat disimpulkan bahwa pada penelitian tersebut PLC dan sensor *proximity* kapasitif dapat digunakan untuk membedakan jenis benda. Keberhasilan pada penelitian Oladapo tersebut mendasari peneliti untuk menciptakan alat serupa dan berguna untuk membedakan jenis sampah organik maupun anorganik.

Pada tahun 2017, Nuraini N.U. melakukan sebuah penelitian dengan menciptakan sebuah sistem klasifikasi sampah organik dan anorganik pada rumah tangga dengan menggunakan metode *Naive Bayes*. Penelitian tersebut menggunakan 3 fitur dalam klasifikasi dan ditentukan oleh 3 sensor yang berbeda. Fitur ke-1 pada penelitian tersebut yaitu data yang diperoleh dari sensor yang serupa pada penelitian Oladapo yakni sensor *proximity* kapasitif. Sensor *proximity* kapasitif tersebut bertujuan untuk mendeteksi kandungan air yang ada pada objek. Fitur 2 didapatkan dari sensor *proximity* induktif yang bertujuan untuk mendeteksi kandungan kayu dan logam. Sedangkan fitur 3 didapatkan dari pembacaan sensor LDR untuk mengetahui tingkat transparansi pada objek yang dideteksi. Pada penelitian tersebut, tingkat keakuratan sensor kapasitif/fitur 1 sebesar 77,5%, tingkat keakuratan sensor induktif/fitur 2 sebesar 100%, tingkat keakuratan sensor LDR/fitur 3 sebesar 100%. Sensor-sensor yang digunakan pada penelitian Nuraini (2017) memiliki akurasi yang cukup baik dan dapat dijadikan acuan dalam membedakan jenis sampah, sehingga pada penelitian ini peneliti memutuskan untuk menggunakan sensor-sensor tersebut.

Mesin yang diciptakan pada penelitian Oladapo (2016) mampu membedakan beberapa jenis benda yang berbeda hanya dengan menggunakan logika matematika sederhana. Sedangkan alat yang diciptakan pada penelitian Nuraini (2017) mampu membedakan jenis sampah menggunakan pengenalan pola *Naive Bayes* dengan data uji sebanyak 20 data dan jumlah data latih sebanyak 160 data, hasil persentase akurasi pada penelitian tersebut berkisar antara 87,50-95,00% (Nuraini, 2017). Pada penelitian ini alat yang diciptakan tidak

menggunakan logika matematika sederhana, melainkan memakai pengenalan pola dengan metode jaringan saraf tiruan *backpropagation*.

Jika pada penelitian Oladapo (2016) menggunakan PLC sebagai pemrosesan datanya dan lebih terfokus pada sektor industri, berbeda dengan penelitian Nuraini (2017) yang menggunakan mikrokontroler Arduino Uno sebagai pemrosesan data dari sensor serta terfokus pada sampah rumah tangga. Namun hasil akuisisi data dari sensor-sensor pada penelitian Nuraini (2017) tidak secara langsung dilakukan klasifikasi pada mikrokontroler yang digunakan, melainkan dikirimkan secara serial dari Arduino Uno menuju perangkat komputer. Klasifikasi dilakukan oleh program yang dijalankan pada komputer dan menampilkan hasilnya pada *Command Prompt*. Pada penelitian ini mikrokontroler Arduino Uno tidak hanya berfungsi untuk mengakuisisi data dari sensor namun juga dapat melakukan klasifikasi, sehingga tidak perlu adanya pengiriman data sensor menuju perangkat komputer. Perangkat komputer tetap dibutuhkan, namun hanya berfungsi sebagai alat pelatihan metode *backpropagation*.

Penelitian selanjutnya dilakukan oleh Prasetya, R. (2017) menggunakan metode jaringan saraf tiruan *backpropagation* untuk memprediksi ketinggian air di kota Bogor. Pada penelitian tersebut menggunakan data curah hujan, penguapan, serta ketinggian air berdasarkan pengamatan tahun 2009-2010. Penelitian tersebut melakukan perancangan model prediksi menggunakan jaringan saraf tiruan *backpropagation* dengan *software* MATLAB. Jumlah data yang digunakan pada penelitian tersebut yaitu sebanyak 395 data yang kemudian setiap data ditransformasi ke dalam kisaran nilai 0,001 sampai dengan 0,098. Komposisi yang diterapkan pada penelitian tersebut yaitu data pelatihan sebanyak 80% sedangkan untuk data pengujian dan validasi sebanyak 20%. Arsitektur jaringan yang digunakan yaitu 2 neuron (penguapan dan curah hujan) pada lapisan masukan, 1 lapisan tersembunyi, serta 1 lapisan keluaran dengan 1 unit neuron (ketinggian air), dengan nilai laju pembelajaran 0,9 dan maksimum *epoch* sebesar 10000. Berdasarkan 10 kali pengujian yang dilakukan oleh Prasetya, R, terdapat *error* sebesar 10,93% sehingga dapat disimpulkan bahwa arsitektur jaringan saraf tiruan yang diterapkan oleh penelitian dapat digunakan untuk memprediksi tingkat ketinggian air pada daerah yang dialiri sungai (Prasetya, 2017). Pada penelitian Prasetya (2017) tersebut hanya terfokus pada perangkat lunak serta tidak terfokus pada klasifikasi benda berdasarkan jenisnya. Penelitian tersebut berhasil melakukan prediksi dengan menggunakan arsitektur jaringan saraf tiruan yang multi-lapis dengan satu lapis tersembunyi. Berdasarkan penelitian tersebut, pada penelitian ini akan digunakan satu lapisan tersembunyi serta. Nilai *epoch* dan nilai laju pembelajaran yang digunakan pada penelitian tersebut dapat dijadikan acuan untuk mencari kombinasi nilai yang mampu menghasilkan *error* terkecil pada penelitian ini.

2.2 Dasar Teori

Berikut merupakan kumpulan dasar-dasar teori yang diterapkan dalam proses implementasi sistem.

2.2.1 Sampah

Sampah adalah bahan yang tidak mempunyai nilai atau tidak berharga dalam pembuatan atau pemakaian barang rusak atau cacat dalam pembuatan manufaktur (Djohan & Halim, 2013). Kamus Besar Bahasa Indonesia (KBBI) versi *online* menyebutkan bahwa sampah merupakan barang atau benda yang sudah dibuang karena pemilik sudah tidak memakai/memerlukannya lagi, misalnya kotoran seperti daun, kertas (Setiawan, 2017).

Berdasarkan berbagai definisi di atas melalui berbagai media, penulis menyimpulkan bahwa sampah merupakan benda yang tidak mempunyai nilai guna dan tidak dapat dimanfaatkan kembali yang dibuang oleh pemiliknya, sehingga sudah tidak mempunyai nilai ekonomis kembali. Seperti yang dapat diamati di sekitar, setiap hari manusia pasti menghasilkan sampah yang tidak sedikit. Sampah-sampah tersebut akan menjadi masalah jika tidak dikelola dengan baik, untuk itu maka dibutuhkan sistem pemilahan sampah yang baik pula agar mempermudah proses penggunaan kembali sampah yang sulit terurai.

2.2.2 Klasifikasi Sampah

Berdasarkan sifatnya, sampah dapat diklasifikasikan menjadi 2 jenis, yaitu sampah anorganik dan organik.

2.2.2.1 Sampah Anorganik

Menurut Djohan & Halim (2013), sampah organik merupakan sampah yang mempunyai sifat *onbiodegradable* sehingga tidak mudah membusuk atau diuraikan secara sempurna melalui proses biologi (aerob atau anaerob). Sampah jenis ini dapat mempunyai nilai ekonomi kembali jika dijadikan produk secara kreatif dan inovatif. Sampah ini dapat dikenali dari kandungan logam untuk sampah anorganik yang berjenis logam dan dari tingkat transparansi cahaya untuk sampah dengan jenis plastik yang tipis.

2.2.2.2 Sampah Anorganik

Menurut Djohan & Halim (2013), sampah organik merupakan kebalikan dari sampah anorganik, yaitu sampah yang mempunyai sifat *biodegradable* sehingga mudah membusuk karena adanya pengurai dan atau sampah yang dapat didegradasi atau diuraikan secara sempurna melalui proses biologi (aerob atau anaerob). Sampah ini dapat dikenali dari kandungan air yang ada hampir di seluruh sampah dengan jenis organik.

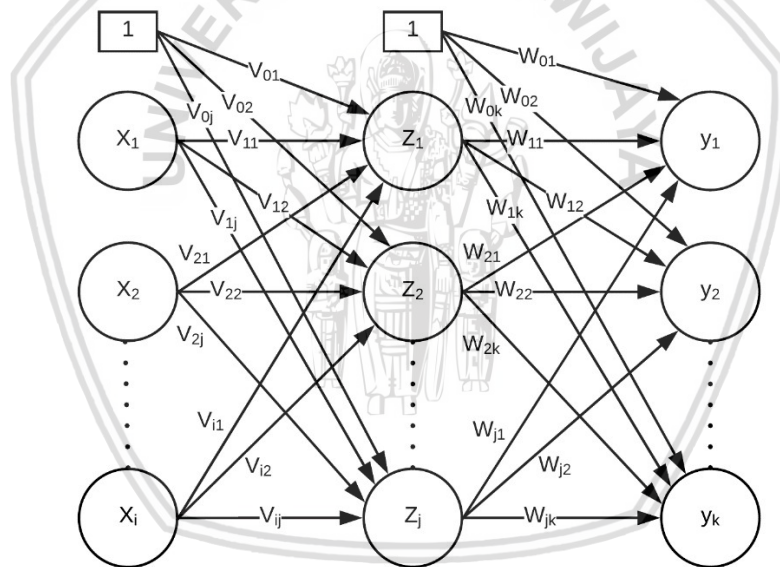
2.2.3 Pengenalan Pola

Pengenalan pola bertujuan untuk membedakan suatu objek dengan objek berdasarkan ciri-ciri yang dimiliki serta menentukan kelompok atau kategorinya berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Pola merupakan entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama, pola dapat berupa kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks (Munir, 2004)

2.2.4 Jaringan Saraf Tiruan

Jaringan saraf tiruan atau dapat disingkat JST merupakan sebuah sistem komputasi yang arsitektur serta operasinya menyerupai arsitektur dan cara kerja sel saraf biologi pada otak. Jaringan saraf tiruan dapat dijadikan sebagai model yang matematis dan komputasi untuk melakukan klasifikasi data, fungsi aproksimasi non-linier, regresi non parametrik, *cluster* atau juga dapat digunakan sebagai simulasi untuk model saraf biologi (Kristanto, 2004).

2.2.5 Backpropagation



Gambar 2.1 Arsitektur jaringan saraf tiruan *backpropagation*
(Mulyanto, Suhartono, & Sutojo, 2011)

Backpropagation adalah suatu model jaringan saraf tiruan yang memiliki topologi multilapis (*multilayer*) yang terdiri dari satu lapisan masukan (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layer*), serta satu lapisan keluaran (*output layer*). Setiap lapis pada jaringan memiliki unit-unit neuron atau *node* yang biasanya dimodelkan dengan bentuk lingkaran. Setiap neuron pada satu lapisan dengan neuron pada lapisan berikutnya saling dihubungkan dan setiap sambungannya memiliki bobot (*weights*), dan bias. Lapisan tersembunyi dapat memiliki bias, yang mana merupakan bobot yang bernilai satu. *Backpropagation* mampu melatih jaringan hingga dapat mengenali pola pada saat pelatihan dan

mampu memberikan respons yang benar terhadap masukan yang serupa (tidak sama) dengan pola yang ada pada data latih (Puspitaningrum, 2006).

2.2.6 Pelatihan Backpropagation

Pelatihan *backpropagation* memiliki 3 fase, yaitu fase *feed forward*, *backpropagation*, dan fase perubahan bobot. Berikut merupakan algoritme dari *backpropagation* (Mulyanto, Suhartono, & Sutojo, 2011):

1. Inisialisasi *epoch*, jumlah unit pada lapisan masukan, tersembunyi, serta lapisan keluaran, inisialisasi *learning rate*, inisialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 3-8.

FASE 1: Propagasi maju (*Feed Forward*)

3. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya
4. Menghitung hasil pada unit tersembunyi z_j ($j=1,2,...,p$)

$$z_{net_j} = v_{jo} + \sum_{i=1}^n x_i v_{ji} \quad (2.1)$$

Menghitung aktivasi dengan fungsi sigmoid:

$$z_j = f(z_{net_j}) = \frac{1}{1+e^{-z_{net_j}}} \quad (2.2)$$

Keluaran dari fungsi aktivasi tersebut dikirim ke semua unit lapisan tersembunyi.

5. Menghitung hasil pada unit keluaran y_k ($k=1,2,...,m$).

$$y_{net_k} = w_{ko} + \sum_{j=1}^p z_j w_{kj} \quad (2.3)$$

Hitung sinyal keluaran menggunakan fungsi aktivasi sigmoid:

$$y_k = f(y_{net_k}) = \frac{1}{1+e^{-y_{net_k}}} \quad (2.4)$$

FASE 2: Propagasi mundur (*Backpropagation*)

6. Setiap unit keluaran Y_k ($k=1,2,...,p$), menerima target pola yang berhubungan dengan pola masukan pelatihan, hitung informasi kesalahannya (*error*):

$$\delta_k = (t_k - y_k) \cdot f'(y_{net_k}) = (t_k - y_k) \cdot y_k(1 - y_k) \quad (2.5)$$

Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki bobot baru)

$$\Delta w_{kj} = \alpha \delta_k z_j \quad (2.6)$$

$$(k=1,2,...,m ; j = 0,1,2,...,p)$$

7. Hitung faktor δ pada unit tersembunyi berdasarkan kesalahan pada unit tersembunyi z_j

$$\delta_{net_j} = \sum_{k=1}^m \delta_j w_{kj} \quad (2.7)$$

Faktor δ unit tersembunyi:

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j(1 - z_j) \quad (2.8)$$

Hitung suku perubahan bobot v_{ji} (yang akan dipakai nanti untuk mengubah bobot v_{ji})

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (2.9)$$

($j = 1, 2, \dots, p$; $i = 0, 1, 2, \dots, n$)

FASE 3: Perubahan bobot

8. Hitung semua perubahan bobot

Perubahan bobot yang menuju unit keluaran:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (2.10)$$

($k=1, 2, \dots, m$; $j = 0, 1, 2, \dots, p$)

Perubahan bobot yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (2.11)$$

($j = 1, 2, \dots, p$; $i = 0, 1, 2, \dots, n$)

9. Uji kondisi berhenti, jika sudah mencapai batas kesalahan yang diharapkan atau batas iterasi maksimal.

2.2.7 Pengukuran Kesalahan dari Perubahan Bobot

Setiap perubahan bobot yang dilakukan pada setiap iterasi akan menghasilkan tingkat kesalahan yang terus menurun. Hal ini dapat diketahui dengan menghitung tingkat kesalahan prediksi menggunakan *Least Square Method* (LSM). Metode ini menjelaskan estimasi dari parameter yang digunakan sebagai nilai yang meminimalkan total dari kuadrat (hingga disebut *least squares*) antara hasil pengukuran dengan nilai hasil prediksi. Berikut merupakan formula untuk mengetahui nilai kesalahan dengan metode *Least Square* (Abdi, 2007), dimana \mathcal{E} merupakan kesalahan / kuantitas yang harus diminimalkan, Y merupakan nilai hasil pengukuran, dan \hat{Y} adalah nilai hasil dari perhitungan / prediksi yang dilakukan.

$$\mathcal{E} = \sum_i (Y_i - \hat{Y}_i)^2 \quad (2.12)$$

2.2.8 Pengukuran Kinerja Klasifikasi

Sebuah sistem klasifikasi tidak akan selalu berhasil dalam setiap klasifikasinya terhadap data uji, akurasi yang dihasilkan tidak selalu dapat meraih 100%. Sebuah sistem akan dikatakan lebih baik bila mampu melakukan klasifikasi dengan akurasi yang cukup tinggi atau bahkan mendekati 100%. Untuk itu, perlu adanya pengukuran akurasi dari klasifikasi yang dilakukan oleh sistem. Pengukuran akurasi ini dapat dilakukan dengan menggunakan matriks konfusi yang berupa Tabel pencatat hasil kerja klasifikasi. Pada Tabel 2.1 menunjukkan matriks konfusi dengan klasifikasi 2 kelas yaitu kelas 0 dan 1, dimana setiap f_{ij} menyatakan jumlah data dari kelas i terhadap j (Prasetyo, 2012).

Tabel 2.1 Matriks konfusi 2 kelas

f_{ij}		kelas prediksi (j)	
		kelas 0	kelas 1
kelas asli (i)	kelas 0	f_{00}	f_{01}
	kelas 1	f_{10}	f_{11}

Matriks konfusi dapat memberi tahu banyaknya data yang diprediksi dengan benar dari masing-masing kelas ($f_{00} + f_{11}$) dan banyaknya kesalahan klasifikasi ($f_{01} + f_{10}$). Sehingga dari Tabel tersebut dapat dihitung tingkat akurasi / seberapa tepat kemampuan prediksi dari sistem klasifikasi tersebut dengan menggunakan persamaan:

$$Akurasi = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \quad (2.13)$$

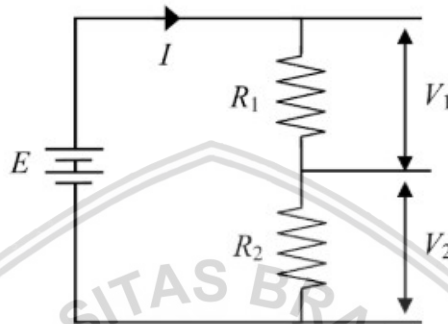
2.2.9 Rata-rata (*average*)

Rata-rata (*average*) merupakan nilai yang dapat mewakili sekelompok data atau himpunan (*a set of data*). Jika pada sekumpulan data yang disusun berdasarkan besar/kecilnya nilai, maka nilai rata-rata pada umumnya cenderung terletak di tengah kelompok data tersebut, sehingga dapat disebut juga ukuran kecenderungan memusat (*measures of central tendency*). Salah satu jenis rata-rata yang sering dipergunakan ialah rata-rata hitung (*mean*). Dalam penelitian ini rata-rata hitung yang digunakan adalah rata-rata perkiraan (sampel) karena dari sekian banyak percobaan hanya akan diambil beberapa sampel untuk dilakukan pengujian. Rata-rata perkiraan (\bar{x}) yakni nilai rata-rata yang dihitung berdasarkan sampel sebanyak n , dimana $n <$ banyaknya observasi yang telah dilakukan (Supranto, 2008).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i \quad (2.14)$$

2.2.10 Rangkaian Pembagi Tegangan

Rangkaian pembagi tegangan digunakan untuk membagi tegangan sumber atau masukan menjadi tegangan keluaran. Pembagi tegangan dapat dilakukan dengan menggunakan dua atau lebih resistor yang dirangkai secara seri. Rangkaian tersebut dijelaskan pada Gambar 2.2 dimana I merupakan arus yang mengalir melalui resistor R_1 dan R_2 , sedangkan E merupakan tegangan masukan yang kemudian akan terbagi menjadi dua bagian keluaran V_1 dan V_2 (Yohandri & Asrizal, 2016).



Gambar 2.2 Rangkaian pembagi tegangan
Sumber: (Yohandri & Asrizal, 2016)

Besarnya tegangan setiap resistor dapat dihitung menggunakan hukum Ohm dimana $V = IR$, sehingga:

$$V_1 = IR_1 \text{ dan } V_2 = IR_2 \quad (2.15)$$

Besarnya I yang mengalir dalam rangkaian ditentukan oleh tegangan sumber dan total resistansi dari resistor.

$$I = \frac{E}{R_1 + R_2} \quad (2.16)$$

Dengan melakukan substitusi pada Persamaan (2.16) kepada Persamaan (2.15), maka dapat diketahui besarnya tegangan dari setiap resistor yang digunakan dengan Persamaan (2.17) dan (2.18).

$$V_1 = \frac{R_1}{R_1 + R_2} \times E \quad (2.17)$$

dan

$$V_2 = \frac{R_2}{R_1 + R_2} \times E \quad (2.18)$$

2.2.11 ADC (*Analog to Digital Converter*)

ADC digunakan untuk mengonversikan sinyal analog menjadi nilai digital. Saat ADC digunakan untuk melakukan pengukuran besaran fisis, salah satu penentu akurasi hasil pengukuran yaitu resolusi ADC. Resolusi ADC tersebut terkait dengan banyaknya bit keluaran pada ADC. Resolusi dapat diartikan juga

sebagai besarnya tegangan yang dibutuhkan untuk setiap langkah naiknya nilai ADC (*step size*). Dengan asumsi bahwa nilai keluaran ADC merupakan nilai diskrit yang dibulatkan ke bawah, maka nilai digital data diketahui dengan persamaan di bawah ini (Dharmawan, 2017).

$$\text{Nilai digital} = \frac{V_{IN} \times 2^n}{V_{REF}} \quad (2.19)$$

Sedangkan untuk resolusi ADC dapat diketahui melalui persamaan di bawah ini (Dharmawan, 2017).

$$\text{Resolusi ADC} = \frac{V_{REF}}{2^n} \quad (2.20)$$

Berdasarkan Persamaan (2.19) dan (2.20) maka dapat diketahui bahwa Persamaan (2.19) ekuivalen dengan persamaan berikut:

$$\text{Nilai digit} = \frac{V_{IN}}{\text{Resolusi ADC}} \quad (2.21)$$

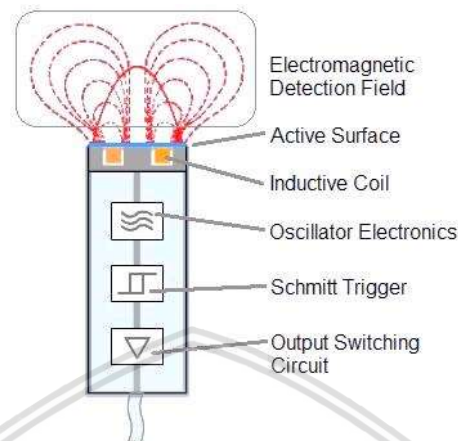
2.2.12 Light Dependent Resistor (LDR)

Light Dependent Resistor atau LDR merupakan salah satu jenis resistor yang memiliki nilai resistansi tergantung pada jumlah cahaya yang mengenai permukaannya. Resistor ini bekerja berdasarkan prinsip konduktivitas foto (cahaya). Konduktivitas material akan berkurang dan juga elektron-elektron dalam pita valensi perangkat tersebut tertarik pada pita konduksi saat cahaya jatuh ke permukaannya. Foton-foton dalam cahaya tersebut harus memiliki energi yang lebih besar dari celah pita dari bahan semikonduktor, sehingga elektron dapat melompat dari pita valensi ke konduksi. Semakin terang cahaya yang menyinari permukaan dari LDR maka resistansi pada LDR akan berkurang, sebaliknya jika cahaya yang menyinari permukaan LDR berkurang, resistansi LDR akan meningkat. (Agarwal, 2015)

2.2.13 Sensor Proximity Induktif

Sensor *proximity* induktif merupakan osilator elektronik yang terdiri dari kumparan induktif yang terbuat dari banyak lilitan kawat tembaga, sebuah kapasitor untuk menyimpan muatan listrik. Ukuran kumparan induktif dan kapasitor disesuaikan agar menghasilkan osilasi gelombang sinus pada frekuensi yang tetap. Osilasi tersebut menghasilkan medan elektromagnetik pada permukaan sensor karena koil induktif berada tepat di belakang permukaan aktif. Ketika sebuah logam konduktif memasuki zona yang ditentukan oleh batas medan elektromagnetik, sebagian energi osilasi ditransfer ke logam target sehingga membentuk sebuah arus listrik kecil yang bersirkulasi yang disebut arus *eddy*. Inilah sebabnya mengapa proksi induktif kadang-kadang disebut sensor arus *eddy*. Arus *eddy* yang mengalir menemui hambatan listrik saat bersirkulasi menciptakan sejumlah kecil daya yang hilang dalam bentuk panas. Kehilangan daya tidak sepenuhnya digantikan oleh sumber energi internal sensor, sehingga intensitas osilasi sensor berkurang. Pemicu *Schmitt* akan mendeteksi bahwa tingkat osilasi

telah turun di bawah tingkat yang telah ditentukan. Batas tingkatan tersebut merupakan suatu kondisi dimana kehadiran target logam telah dipastikan. Setelah pendeteksian target oleh pemicu *Schmitt*, keluaran sensor diaktifkan (Menke, 2014)



Gambar 2.3 Elemen dalam sensor *proximity* induktif (Menke, 2014)

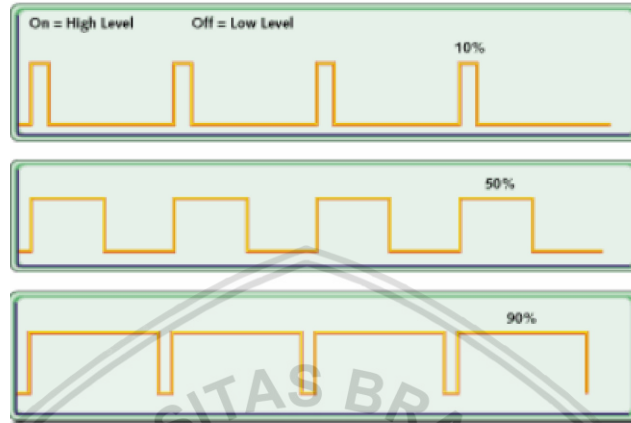
2.2.14 Sensor *Proximity* Kapasitif

Sensor kapasitif mampu mengukur perubahan kapasitans yang terjadi. Kapasitans menggambarkan bagaimana dua benda konduktif dengan jarak di antara keduanya dapat merespons perbedaan voltase yang diterapkan padanya. Tegangan yang diterapkan pada konduktor menghasilkan medan listrik di antara kedua objek tersebut, sehingga muatan positif dan negatif akan saling menarik pada setiap objek. Jika polaritas tegangan terbalik, maka muatan juga akan terbalik. Sensor kapasitif menggunakan tegangan terbalik yang menyebabkan muatan untuk terus membalikkan posisi mereka. Pergerakan muatan tersebut menciptakan arus listrik bolak-balik yang dideteksi oleh sensor. Banyaknya aliran arus ditentukan oleh kapasitans, dan kapasitans ditentukan oleh luas permukaan dan kedekatan objek konduktif. Objek yang lebih besar dan lebih dekat menyebabkan arus yang lebih besar daripada objek yang lebih kecil dan lebih jauh. Kapasitans juga dipengaruhi oleh jenis bahan non-konduktif di celah antara objek (Kretschmar, 2009).

2.2.15 Pulse Width Modulation (PWM)

Pulse Width Modulation atau PWM merupakan teknik yang umum digunakan untuk mengontrol sirkuit analog atau daya ke perangkat listrik inersia seperti servo dengan memanfaatkan keluaran digital dari prosesor. Kontrol digital digunakan untuk membuat sinyal persegi yang beralih antara nyala dan mati. Durasi pada saat nyala disebut sebagai lebar pulsa (*pulse width*). Jika sebuah prosesor dengan sumber daya sebesar 5V memberikan sinyal persegi dengan siklus tugas (*duty cycle*) 50%, maka tegangan yang dihasilkan yaitu 2,5V. Siklus tugas lainnya dapat menghasilkan tegangan antara 0-100% dari sumber daya yang

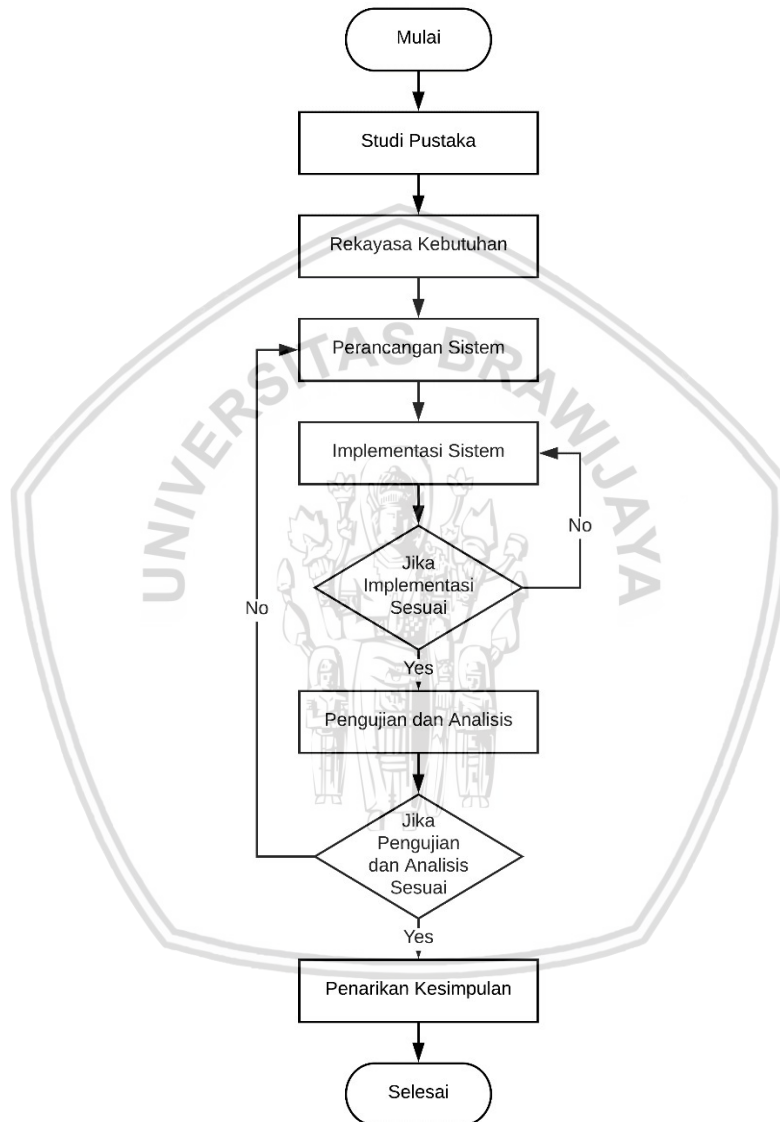
digunakan tergantung pada resolusi PWM. Siklus tugas didefinisikan sebagai persentase dari sinyal digital tinggi ke digital rendah selama periode PWM. Resolusi PWM merupakan jumlah maksimum pulsa yang dapat dimasukkan dalam periode PWM. Periode PWM yaitu jangka waktu pada saat PWM terjadi (Hui, 2012). Gambar 2.4 menunjukkan perbedaan bentuk sinyal PWM dengan siklus tugas 10%, 50%, dan 90%.



**Gambar 2.4 Sinyal PWM dengan siklus tugas yang bervariasi
(Hui, 2012)**

BAB 3 METODOLOGI PENELITIAN

Pada bab ini dibahas mengenai tahapan-tahapan yang dilakukan dalam penelitian ini. Tahapan tersebut meliputi studi dan pengkajian pustaka, rekayasa kebutuhan sistem, perancangan sistem, implementasi sistem, tahap pengujian dan analisis, serta tahap penarikan kesimpulan.



Gambar 3.1 Diagram alir metode penelitian

Diagram alir pada Gambar 3.1 di atas menggambarkan alur selama proses pengerjaan skripsi. Ada beberapa tahap yang memerlukan syarat, seperti setelah melakukan implementasi sistem jika tidak sesuai maka harus kembali ke tahap implementasi sistem hingga selesai. Serta ada syarat setelah melakukan pengujian dan analisis, saat pengujian dan analisis tidak sesuai maka harus kembali ke tahap perancangan sistem.

3.1 Studi Pustaka

Tahap studi pustaka merupakan tahap pertama yang ditempuh dengan tujuan menambah studi pustaka serta menambah pengetahuan yang berkaitan dan mendukung proses penelitian ini. Tahap ini dilakukan dengan mengumpulkan teori dan pustaka yang berkaitan dengan penelitian ini meliputi:

1. Perbedaan sampah organik dan sampah anorganik.
2. Algoritme jaringan saraf tiruan *backpropagation*.
3. Pelatihan data dan mendapatkan bobot dengan tingkat kesalahan yang rendah.
4. Pengukuran kinerja klasifikasi yang dilakukan oleh sistem.
5. Perhitungan rata-rata hitung.
6. Penggunaan rangkaian tegangan.
7. Melakukan konversi sinyal analog dari sensor-sensor yang digunakan menjadi data digital.

3.2 Rekayasa Kebutuhan

Tahap rekayasa kebutuhan ini bertujuan untuk menganalisa kebutuhan-kebutuhan yang mendukung dalam proses penelitian ini. Untuk dapat menentukan kebutuhan, maka perlu adanya gambaran umum terhadap sistem yang akan dibangun dimana membahas mengenai perspektif sistem, ruang lingkup, serta karakteristik pengguna. Setelah itu barulah dilakukan analisis kebutuhan sistem meliputi analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional. Analisis non-fungsional meliputi analisis kebutuhan perangkat keras dan kebutuhan perangkat lunak. Berikut merupakan perangkat keras yang dibutuhkan dalam penelitian ini antara lain:

1. Perangkat keras mampu membaca nilai dari sensor analog dan memiliki jumlah pin yang cukup untuk membaca nilai pada masing-masing sensor.
2. Perangkat keras mampu mengontrol aktuator menggunakan sinyal PWM (*Pulse Width Modulation*).
3. Perangkat keras dapat diprogram dengan melakukan *upload* untuk dapat membaca nilai sensor yang digunakan, serta dapat mengontrol aktuator.
4. Perangkat keras harus mempunyai kapasitas memori yang cukup untuk menyimpan bobot yang didapatkan dari hasil latih, serta memiliki kemampuan proses yang mumpuni untuk melakukan perhitungan dengan metode jaringan saraf tiruan.

Sedangkan perangkat lunak yang dibutuhkan dalam penelitian ini antara lain:

1. Perangkat lunak yang dapat digunakan untuk membuat program yang dapat dipasang di mikrokontroler.
2. Perangkat lunak yang dapat menyimpan dan mengolah basis data yang dapat diproses oleh program.

Terakhir yaitu penentuan batasan sistem berdasarkan analisis kebutuhan non-fungsional maupun fungsional yang telah ditentukan.

3.3 Perancangan Sistem

Perancangan sistem ini dilakukan setelah terpenuhinya analisis kebutuhan pada tahap sebelumnya. Perancangan sistem pada penelitian ini mendefinisikan sistem apa yang akan peneliti bangun sehingga akan lebih terstruktur dan sistematis. Perancangan sistem ini meliputi perancangan perangkat keras, perancangan perangkat lunak, serta perancangan jaringan saraf tiruan. Perancangan perangkat keras pada penelitian ini dijelaskan pada poin-poin berikut:

1. Data dari sampah yang akan diklasifikasi diakuisisi oleh 3 buah sensor, yaitu sensor *proximity* kapasitif untuk membaca kandungan air pada objek tersebut, *proximity* induktif untuk membaca kandungan logam, dan sensor LDR dengan LED bertugas untuk membaca banyaknya cahaya dapat melewati objek tersebut (transparansi).
2. Data yang diterima oleh sensor kemudian diolah dengan metode jaringan saraf tiruan untuk memprediksi jenis sampah berdasarkan sampah yang terdeteksi oleh sensor.
3. Ketika hasil prediksi menunjukkan bahwa sampah tersebut adalah anorganik maka servo akan berputar 90° searah jarum jam, sedangkan jika hasil prediksi menunjukkan organik maka servo akan berputar 90° berlawanan arah jarum jam sehingga tempat sampah dapat dikelompokkan sesuai pada tempatnya.

Sedangkan untuk perancangan perangkat lunak yaitu bagaimana program dapat membaca data latih dari basis data, bagaimana melakukan pelatihan dan dapat menginformasi hasil perubahan bobot terakhir dari hasil pelatihan. Pada perancangan jaringan saraf tiruan akan dibahas tentang perancangan arsitektur, pengumpulan data latih, serta pelatihannya.

3.4 Implementasi Sistem

Tahap implementasi sistem ini dilakukan berdasarkan tahap perancangan yang telah ditentukan sebelumnya. Pada tahap ini gagasan desain maupun hitungan yang merupakan tujuan sebelumnya akan direalisasikan. Berikut merupakan beberapa tahap implementasi pada penelitian ini yaitu:

1. Implementasi perangkat keras, dimana sistem ini terdiri dari beberapa alat berdasarkan kebutuhan perangkat keras yang telah ditentukan sesuai perancangan sistem yang dibuat.
2. Implementasi perangkat lunak, dimana program yang diterapkan pada mikrokontroler dapat menjalankan sistem sesuai keinginan, serta program yang akan diterapkan pada komputer dapat melakukan pelatihan dengan

metode jaringan saraf tiruan *backpropagation* dan menampilkan seluruh bobot hasil pelatihan.

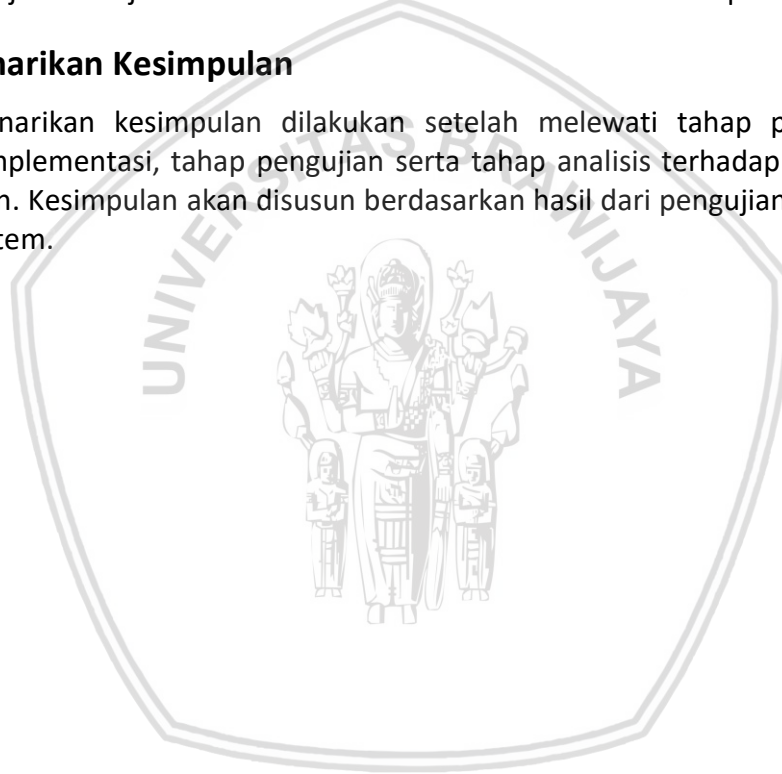
3.5 Pengujian dan Analisis

Tahap pengujian dan analisis ini bertujuan untuk mengetahui kesesuaian sistem setelah melakukan proses implementasi. Kesesuaian tersebut dapat diketahui dengan membandingkannya dengan kebutuhan sistem pada tahap perancangan. Berikut merupakan pengujian yang dilakukan yakni:

1. Pengujian fungsionalitas setiap sensor yang digunakan.
2. Pengujian akurasi sistem dalam melakukan prediksi menggunakan metode jaringan saraf tiruan berdasarkan bobot yang diperoleh dari hasil pelatihan.
3. Pengujian kinerja dalam melatih data serta dalam melakukan prediksi.

3.6 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan setelah melewati tahap perancangan, tahap implementasi, tahap pengujian serta tahap analisis terhadap sistem yang dibangun. Kesimpulan akan disusun berdasarkan hasil dari pengujian dan analisis pada sistem.



BAB 4 REKAYASA KEBUTUHAN

Bab ini menjelaskan tentang kebutuhan-kebutuhan yang harus dipenuhi dalam melakukan perancangan sampai dengan tahap implementasi. Dengan adanya rekayasa kebutuhan diharapkan sistem yang akan dibuat dapat bekerja dengan baik.

4.1 Gambaran Umum Sistem

Sistem ini bekerja untuk mempermudah pengguna dalam mengelompokkan sampah yang ingin dibuangnya menjadi sampah organik maupun anorganik. Sistem ini memiliki bagian utama berbentuk kubus dimana di atasnya terdapat 3 buah sensor, yaitu sensor LDR, sensor *proximity*. Saat pengguna hendak membuang sampah, pengguna perlu meletakkannya di atas alat, sehingga sampah tersebut akan terdeteksi oleh sensor-sensor yang ada pada sistem. Sistem akan membaca tingkat transparansi, induktansi, serta kapasitansi dari benda tersebut. Data-data tersebut akan diolah dan diklasifikasikan berdasarkan jenis sampahnya dengan metode jaringan saraf tiruan. Jika hasil klasifikasi menunjukkan hasil organik maka sistem akan membuka tutup tempat sampah pada bagian sampah organik, namun jika hasil klasifikasi menunjukkan hasil anorganik maka tutup pada bagian sampah anorganik akan terbuka.

4.1.1 Perspektif Sistem

Sistem ini dikatakan berjalan dengan semestinya apabila sesuai dengan yang diharapkan yaitu sistem mampu membaca tingkat transparansi, kandungan logam, dan kadar air pada benda yang dideteksi, serta sistem mampu melakukan klasifikasi terhadap benda yang terdeteksi.

4.1.2 Ruang Lingkup

Alat ini akan dipasang seperti layaknya tempat sampah pada rumah/perkantoran pada umumnya. Sistem ini memiliki sensor yang tertanam pada tempat sampah tersebut, ketika jenis sampah sudah terdeteksi maka tutup dari tempat sampah tersebut akan berputar untuk membuka ruang jenis sampah yang sejenis dengan sampah yang terdeteksi.

4.1.3 Karakteristik Pengguna

Karakteristik pengguna diperuntukkan bagi masyarakat awam yang tidak mengetahui dan memahami jenis sampah yang akan dibuangnya.

4.2 Analisis Kebutuhan Sistem

Pada subbab ini akan dibahas kebutuhan antarmuka dari sistem yang akan dirancang dengan ketentuan sebagai berikut.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi dalam sebuah sistem. Berikut beberapa kebutuhan fungsional dari sistem yang harus berfungsi agar dapat bekerja dengan baik.

1. Setiap sensor dapat membaca kandungan sampah dan transparansi benda sesuai dengan benda yang dapat dideteksi

Fungsi ini berguna untuk mendapatkan nilai transparansi, kandungan logam, serta nilai kapasitif dari sampah yang di deteksi.

2. Pelatihan data dengan algoritme jaringan saraf tiruan *backpropagation*.

Pada fungsionalitas ini bertujuan untuk mencari bobot dari lapis masukan ke lapis tersembunyi dan dari lapis tersembunyi ke lapis keluaran dengan menggunakan metode jaringan saraf tiruan *backpropagation*. Fungsi tidak dijalankan pada sistem yang dibuat namun berjalan di sistem lain.

3. Menggerakkan servo untuk membuka tutup bagian sampah tertentu.

Fungsi ini berguna untuk memberi tahu hasil dari klasifikasi kepada pengguna, sehingga pengguna hanya perlu langsung memasukkan sampah yang dibawa ke tempat yang sejenis.

4. Prediksi data berdasarkan data yang diterima oleh sensor.

Fungsi ini berguna melakukan prediksi pada mikrokontroler dengan menggunakan bobot-bobot hasil pelatihan.

4.2.2 Analisis Kebutuhan Non-fungsional

Kebutuhan non fungsional pada sistem yang dibangun terdiri kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang digunakan pada sistem ini adalah sebagai berikut:

A. Arduino Uno

Arduino Uno merupakan mikrokontroler yang berfungsi sebagai otak dari sistem yang dapat diberikan sebuah program dengan cara melakukan *upload* melalui sambungan serial USB sehingga dapat bekerja sesuai yang diharapkan. Arduino Uno dipilih karena sudah memiliki ADC (*Analog to Digital Converter*) yang dapat membaca perubahan tegangan yang masuk dari sensor-sensor yang digunakan dan diubah menjadi data digital yang dapat diolah oleh mikrokontroler ATmega328. Setiap pin Analog pada Arduino terhubung dengan ADC, sehingga setiap sensor yang digunakan hanya perlu menghubungkan pin data pada sensor dengan pin analog pada *board* Arduino Uno. Arduino Uno memiliki 6 pin analog,

sedangkan sistem membutuhkan 3 buah sensor analog. ADC pada Arduino Uno memiliki panjang 10bit, sehingga dapat mengubah tegangan dengan rentang 0–5v menjadi data dengan rentang 0–1023. Arduino Uno juga memiliki beberapa pin yang mendukung PWM (*Pulse Width Modulation*) yang dapat digunakan untuk mengirim sinyal digital melalui pin tersebut. Berdasarkan pertimbangan di atas, Arduino Uno dipilih oleh peneliti sebagai mikrokontroler yang akan diterapkan pada penelitian ini.

Tabel 4.1 Spesifikasi Arduino UNO

Nama	Spesifikasi
Mikrokontroler	ATmega328
Tegangan operasional	5V
Batas tegangan masuk	6 – 20V
Pin I/O <i>digital</i>	14 pin (6 pin mendukung PWM)
Pin analog	6 pin
ADC	10 bit
Arus DC setiap pin I/O	40 mA
Arus DC pada pin 3.3V	50 mA
<i>Flash Memory</i>	32 KB (ATmega328)
	0,5 KB untuk <i>bootloader</i>
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Kecepatan <i>Clock</i>	16 MHz

Sumber: (Arduino, 2016)

B. LDR (*Light Dependent Resistor*)

LDR (*Light Dependent Resistor*) adalah sensor cahaya berupa resistor yang dapat memiliki nilai resistansi yang berubah-ubah berdasarkan intensitas cahaya yang diterimanya. Seperti yang telah kita ketahui bersama, beberapa benda yang terbuat dari kaca dan plastik memiliki karakteristik seperti permukaannya dapat ditembus oleh cahaya (transparansi) yang mana bahan dasar keduanya merupakan jenis sampah anorganik. Tingkat transparansi sampah dapat diketahui berdasarkan banyaknya cahaya yang mampu melewati benda tersebut, oleh karena itu sistem membutuhkan sensor LDR yang dapat mendeteksi tingkat transparansi atau cahaya yang dapat melewati sampah tersebut.

Tabel 4.2 Spesifikasi LDR

Nama	Spesifikasi
Tegangan operasional	3-5 V
Jenis	Sensor Analog

C. LED (*Light Emitting Diodes*)

LED (*Light Emitting Diodes*) merupakan suatu semikonduktor yang dapat memancarkan cahaya ketika diberikan tegangan. Dikarenakan setiap sampah yang dideteksi dapat memiliki perbedaan tingkat transparansi, sensor LDR membutuhkan sumber cahaya yang konstan agar hasil klasifikasi dapat lebih akurat. Oleh sebab itu LED diperlukan sebagai sumber cahaya yang konstan untuk menyinari sensor LDR.

D. Sensor *Proximity*

Berdasarkan penelitian yang pernah dilakukan sebelumnya seperti yang tercantum pada tinjauan pustaka, yaitu penelitian yang dilakukan oleh Oladapo (2016) dan Nuraini (2017) membahas bahwa sensor *proximity* dapat digunakan untuk membedakan sampah berdasarkan jenisnya. Sensor induktif dipilih karena dapat mendeteksi kandungan logam pada sampah yang lebih mengarah ke jenis sampah anorganik. Kemudian sensor *proximity* kapasitif, sensor ini dapat mendeteksi kandungan air yang lebih mengarah ke jenis sampah organik.

Tabel 4.3 Spesifikasi *Proximity* Induktif Lj12a3-4-Z

Nama	Spesifikasi
Tegangan operasional	6-36 V
Arus maksimal	300mA
Jarak deteksi	4mm

Tabel 4.4 Spesifikasi *Proximity* Kapasitif CR18-8DP

Nama	Spesifikasi
Tegangan operasional	12-24 V
Arus maksimal	200mA
Jarak deteksi	8mm

E. Modul *Step Up*

Berdasarkan *datasheet* dari kedua jenis sensor *proximity* yang digunakan pada penelitian ini membutuhkan sumber tegangan minimal 6V dan 12V, sedangkan sumber tegangan yang dihasilkan oleh Arduino Uno hanya 3V dan 5V.

Maka butuh sebuah komponen yang mampu memberikan tegangan yang lebih tinggi dari yang dihasilkan oleh Arduino UNO, sehingga sensor *proximity* mampu mendeteksi objek dengan baik.

Tabel 4.5 Spesifikasi modul *Step Up* XL6009E

Nama	Spesifikasi
Tegangan masuk	-0,3 – 36 V
Tegangan keluar	-0,3 – 60 V

F. *Switch* (Saklar)

Switch yaitu salah satu komponen elektronika yang berfungsi sebagai pemutus atau penghubung aliran listrik. Pada penelitian ini *switch* digunakan sebagai pemicu agar sistem melakukan prediksi serta menggerakkan aktuator.

G. Servo

Servo adalah sebuah motor yang dapat mengubah energi elektrik menjadi energi mekanik melalui interaksi dari dua medan magnet. Sistem membutuhkan sebuah keluaran / aktuator yang dapat menginformasikan kepada pengguna hasil dari klasifikasi yang dilakukan oleh sistem. Berdasarkan paparan yang telah dijelaskan sebelumnya, sistem dapat membuka tutup dari tempat sampah yang sesuai dengan jenisnya maka diperlukan sebuah aktuator berupa Servo yang dapat dikontrol arah putaran dan sudut perputarannya menggunakan PWM dari mikrokontroler.

Tabel 4.6 Spesifikasi Servo MG-NFC

Nama	Spesifikasi
Tegangan operasional	4,8 – 6V DC
Torsi	5,5 kg/cm (pada 4,8 V)
Kecepatan	0,22 SEC / 60
Putaran Maksimum	180 derajat

H. Komputer

Komputer merupakan sekelompok alat elektronik yang memiliki perintah *input*, dapat mengolah *input*, serta memiliki peralatan *output* yang dapat memberikan informasi yang dibutuhkan oleh pengguna serta dapat bekerja secara otomatis (Susanto, 2003). Komputasi jaringan saraf tiruan membutuhkan banyak langkah / proses pada saat pelatihan, selain itu membutuhkan cukup banyak kapasitas simpanan yang dapat menampung data latih. Oleh sebab itu mikrokontroler Arduino Uno dikhawatirkan tidak mampu menampung memori untuk data latih sehingga membutuhkan sistem lain yang lebih mumpuni. Hasil dari perhitungan jaringan saraf buatan dapat berupa bobot sebelum melakukan

prediksi terhadap suatu data yang diberikan. Pada penelitian ini komputer digunakan sebagai sistem tambahan untuk mengerjakan proses pelatihan untuk mencari bobot, serta sebagai penyimpanan data latih.

Tabel 4.7 Spesifikasi Komputer

Nama	Spesifikasi
Tipe	Lenovo g400s
Prosesor	Intel Core i3-3110M
Jumlah core	4
Kecepatan Clock	2,40GHz
Sistem Operasi	Windows 10
RAM	6 GB

4.2.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan dalam penelitian ini digunakan untuk membangun algoritme pada mikrokontroler Arduino serta algoritme untuk melakukan pelatihan data.

A. PlatformIO IDE

PlatformIO IDE (*Integrated Development Environment*) merupakan aplikasi yang mendukung pemrograman sistem tertanam seperti Arduino dengan menggunakan bahasa C++. PlatformIO hanya dapat berjalan di atas aplikasi kode *editor*, pada saat ini kode *editor* yang didukung yaitu: Atom dan Visual Code, pada penelitian ini peneliti memilih aplikasi Visual Code karena lebih banyak fitur yang akan didapatkan tanpa perlu pemasangan tambahan seperti yang tertera pada situs penyedia yaitu <https://platformio.org/platformio-ide>. PlatformIO IDE dapat digunakan untuk membangun dan melebur program sehingga dapat dijalankan pada mikrokontroler Arduino. Karena PlatformIO berjalan di atas kode *editor* Visual Code, maka PlatformIO IDE memiliki berbagai fitur tambahan yang sangat membantu dalam pembangunan program, seperti fitur *autocomplete* yang dapat menampilkan daftar prediksi teks yang ingin ditulis sehingga dapat mempersingkat waktu karena tidak harus menulis teks secara keseluruhan, serta fitur *error detection* yang dapat memeriksa kesalahan pada program sebelum membangun kode program tersebut. Pada penelitian ini PlatformIO IDE digunakan untuk membangun program yang akan dijalankan pada mikrokontroler sistem.

B. Library Servo

Library servo berguna untuk mempermudah dalam proses pemrograman servo pada Arduino. *Library* ini sudah mencakup perintah dasar untuk mengendalikan aktuator servo. Dengan bantuan *library* ini, servo dapat dengan

mudah dikendalikan hanya dengan memanggil fungsi-fungsi yang sudah disediakan dalam *library*.

C. PyCharm IDE

PyCharm IDE merupakan IDE yang dapat membantu para pengembang bahasa Python dalam membangun sebuah program. PyCharm dilengkapi dengan fitur-fitur yang serupa dengan PlatformIO IDE seperti *autocomplete*, *error detection*, dan beberapa fitur lainnya dapat mempermudah dan mempercepat pembangunan sebuah program serta meminimalisir kesalahan yang terjadi. Pada penelitian ini PyCharm digunakan untuk membangun program yang berfungsi untuk melakukan klasifikasi dengan metode jaringan saraf tiruan. Program ini akan menghasilkan bobot-bobot yang diperlukan dalam melakukan prediksi data yang akan dijalankan pada Arduino. Program ini tidak dijalankan pada sistem yang dibuat.

D. MySQL

MySQL merupakan perangkat lunak untuk manajemen basis data SQL. Pada penelitian ini MySQL dibutuhkan untuk menyimpan data latih untuk melakukan *training* pada program python.

E. Library MySQLdb

Library MySQLdb merupakan *library* untuk pemrograman bahasa python yang berfungsi untuk mengolah data yang terekam pada basis data MySQL. Pada penelitian ini, *library MySQLdb* digunakan untuk mengakses data latih dari sebuah Tabel yang telah disediakan. Data-data tersebut dapat diambil dalam bentuk *array*.

4.3 Batasan Sistem

Dalam pembangunan sistem ini, terdapat beberapa batasan sehingga dalam perancangan, implementasi dan pembahasannya tidak terlalu luas. Berikut batasan-batasan sistem yang dimaksud:

1. Sistem hanya dapat membaca objek jika *switch/sakelar* diaktifkan.
2. Pengguna harus mematikan *switch/sakelar* setelah memasukkan sampah pada tempatnya.
3. Objek yang dideteksi maksimal berjarak 0,4cm dari sensor *proximity*.
4. Objek yang dideteksi memiliki lebar maksimal 7cm dan minimal 2,5cm serta memiliki tinggi minimal 2cm dan panjang minimal 1,5 cm.
5. Arsitektur jaringan saraf tiruan hanya memiliki 1 lapisan tersembunyi.
6. Pelatihan dan penyimpanan data latih dilakukan pada sistem lain yang lebih mumpuni seperti laptop.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

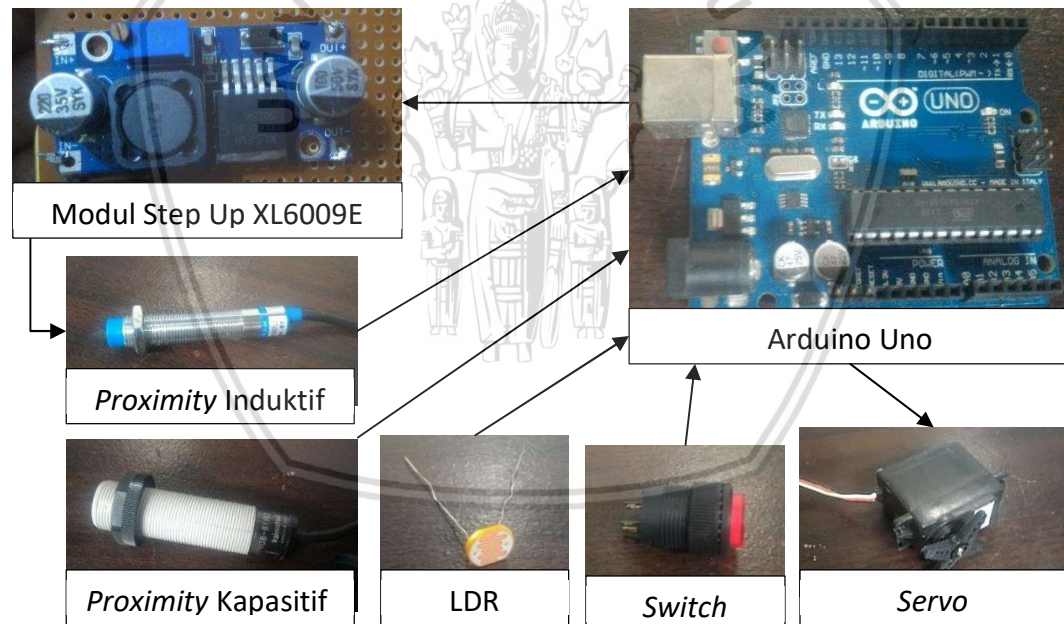
Pada bab ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak untuk dapat melakukan klasifikasi mengenai jenis sampah yang dideteksi.

5.1 Perancangan Sistem

Perancangan sistem ini terdiri dari perancangan perangkat keras, perancangan perangkat lunak, serta perancangan jaringan saraf tiruan..

5.1.1 Perancangan Perangkat keras

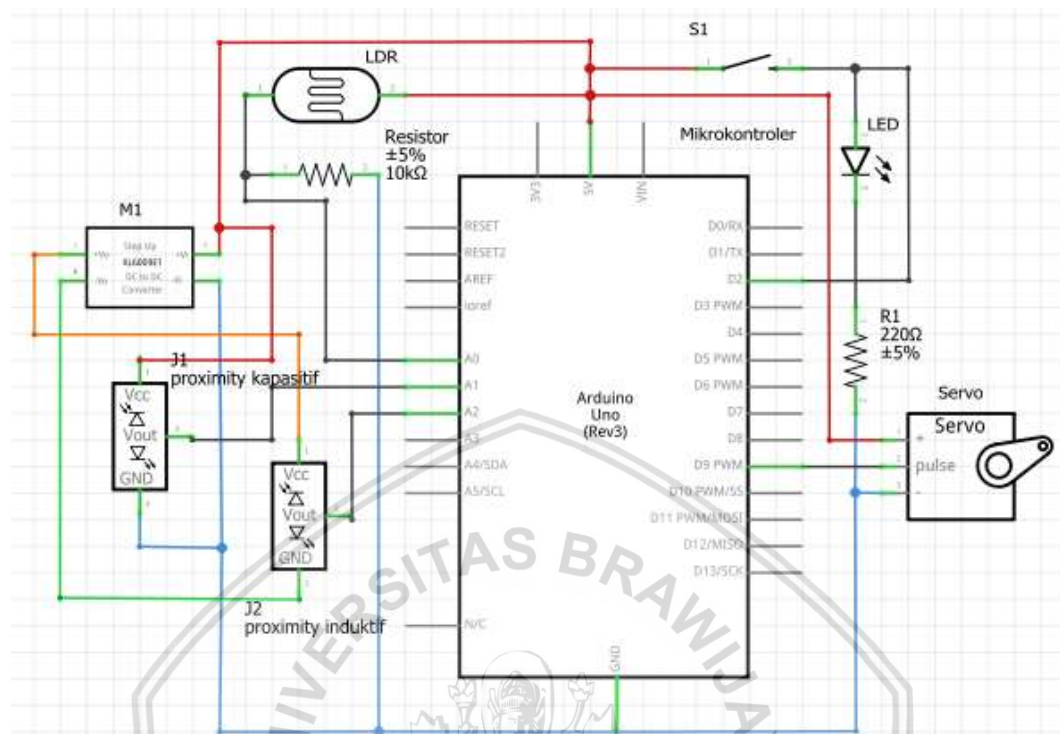
Pada sub bab ini, menjelaskan tentang perancangan sistem dari segi perangkat keras yang meliputi diagram blok, skema, dan desain prototipe. Diagram blok yang diterapkan pada sistem ini digambarkan pada Gambar 5.1. Diagram blok perangkat keras dirancang berdasarkan hasil analisis kebutuhan perangkat keras.



Gambar 5.1 Diagram blok sistem

Sensor *proximity* kapasitif CR18-8DP, *proximity* induktif Lj12a3-4-Z, dan *Light Dependent Resistor* (LDR) merupakan masukan dari sistem yang akan dibaca oleh Mikrokontroler Arduino Uno. Ketiga sensor tersebut merupakan sensor analog akan dibaca oleh mikrokontroler Arduino Uno melalui pin analog. Pada mikrokontroler, sistem akan memproses data yang diterima untuk melakukan

prediksi berdasarkan bobot yang diperoleh dari pelatihan menggunakan program python. Skema rangkaian tersebut dapat dilihat pada Gambar 5.2.



Gambar 5.2 Skema rangkaian sistem

Berdasarkan skema dari Gambar 5.2 terlihat bahwa salah satu kaki sensor LDR terhubung ke pin VCC 5V, sedangkan salah satu kaki lainnya terhubung langsung ke pin A0 dan ke sebuah resistor 10K yang terhubung dengan GND, rangkaian tersebut merupakan rangkaian pembagi tegangan tersebut yang berfungsi untuk mengonversi nilai resistansi dari resistor LDR menjadi tegangan sehingga perubahan tegangan yang terjadi dapat terbaca dengan baik oleh ADC pada Arduino Uno. Karena ADC mendapatkan masukan dari tegangan keluaran (V_{out}) pada rangkaian pembagi tegangan, maka ADC akan membaca tegangan dari V_{out} dan GND (ground).

Berdasarkan hasil pengukuran yang peneliti lakukan menggunakan alat ukur berupa multimeter digital, sensor LDR dapat memberikan resistansi $\pm 2,1 \text{ K}\Omega$ pada saat disinari lampu LED dan pada saat tidak disinari oleh lampu LED resistansi pada LDR meningkat hingga $\pm 230 \text{ K}\Omega$. Berdasarkan pengukuran tersebut, dapat diketahui voltase yang dihasilkan oleh rangkaian pembagi tegangan sistem pada saat sensor LDR disinari oleh LED (V_{out1}) dan saat LDR tidak disinari oleh LED (V_{out2}). Sehingga untuk mengetahui besarnya tegangan yang masuk pada ADC dapat menggunakan Persamaan (2.18). Di bawah ini merupakan contoh perhitungan dari V_{out1} , dimana R_1 adalah resistansi LED saat terang dan diketahui bahwa $R_2 = 10 \text{ K}\Omega$ dan $E = 5 \text{ V}$.

$$R_1 = 2100 \, \Omega$$

$$V_{out1} = \frac{R_1}{R_1 + R_2} \times E$$

$$V_{out1} = \frac{10000}{2100 + 10000} \times 5$$

$$V_{out1} = \frac{10000}{12100} \times 5 = 0.826 \times 5 = 4,132V$$

Berdasarkan perhitungan di atas, maka voltase yang akan dikeluarkan oleh sensor LDR pada saat disinari oleh LED yaitu 4,132V. Kemudian dengan menggunakan Persamaan (2.18) dan hasil pengukuran sebelumnya, peneliti dapat mengetahui voltase yang dihasilkan oleh sensor LDR saat tidak disinari oleh LED berdasarkan perhitungan di bawah ini.

$$R_1 = 230000 \, \Omega$$

$$V_{out2} = \frac{R_2}{R_1 + R_2} \times 5$$

$$V_{out2} = \frac{10000}{230000 + 10000} \times 5$$

$$V_{out2} = \frac{10000}{240000} \times 5 = 0.04 \times 5 = 0.2V$$

Berdasarkan perhitungan di atas, maka dapat disimpulkan bahwa saat LDR tidak disinari oleh LED, tegangan yang dihasilkan LDR sebesar 0,2 Volt sedangkan ketika LDR disinari oleh LED, tegangan yang dihasilkan dapat mencapai 4.132 Volt. Hal ini mendasari peneliti dalam penentuan bit ADC yang akan digunakan. Setiap ADC dengan jumlah bit yang berbeda meskipun dengan tegangan yang sama maka akan memiliki resolusi yang berbeda. Sebagai contoh, jika peneliti menggunakan ADC 10 bit dan tegangan referensi sebesar 5V, maka dengan menggunakan Persamaan (2.20) dapat diketahui besarnya resolusi yang dimiliki ADC tersebut. Berikut contoh perhitungannya:

$$\text{Resolusi ADC} \quad \frac{V_{REF}}{2^n} = \frac{5V}{2^{10}} = \frac{5V}{1024} = 0,004883V$$

Berdasarkan perhitungan di atas maka pada ADC 10bit, setiap peningkatan 0,0049 Volt akan menyebabkan hasil konversi ADC meningkat 1 nilai.

Perhitungan V_{out1} dan V_{out2} yang telah dilakukan dapat digunakan untuk mengetahui nilai hasil pembacaan ADC 10bit pada LDR saat disinari oleh LED maupun tidak dengan menggunakan Persamaan (2.21). Di bawah ini merupakan

contoh perhitungan mendapatkan nilai ADC maksimum dari tegangan keluaran LDR.

$$\text{nilai } ADC_{maks} = \frac{V_{out1}}{\text{Resolusi } ADC}$$

$$\text{nilai } ADC_{maks} = \frac{4,132 \text{ V}}{0,004883 \text{ V}} = 846.28 = 846$$

Sedangkan untuk mengetahui nilai ADC minimum dapat diketahui berdasarkan perhitungan di bawah ini.

$$\text{nilai } ADC_{min} = \frac{V_{out}}{\text{Resolusi } ADC}$$

$$\text{nilai } ADC_{min} = \frac{0,2 \text{ V}}{0,004883 \text{ V}} = 40,96 = 40$$

Maka dapat disimpulkan bahwa ADC akan menghasilkan nilai 846 saat LDR disinari dan 40 saat LDR tidak disinari. Berdasarkan persamaan dan perhitungan-perhitungan yang dilakukan sebelumnya, peneliti mencoba untuk mengetahui nilai dari maksimum dan minimum hasil pembacaan LDR pada beberapa ADC berbeda. Hasil tersebut ditunjukkan pada Tabel 5.1.

Tabel 5.1 Tabel batas nilai minimum dan maksimum nilai ADC berdasarkan perubahan tegangan dari LDR

Bit ADC	Resolusi (Volt)	Nilai minimum	Nilai maksimum	Rentang
2	1.25	0	3	3
4	0.3125	0	13	13
6	0.078125	2	52	50
8	0.01953125	10	211	201
10	0.004882813	40	846	806
12	0.001220703	163	3385	3222

Berdasarkan Tabel 5.1, diketahui rentang pembacaan dari masing-masing ADC. Semakin tinggi rentangnya maka akan semakin presisi, namun apabila ketika pembacaan nilai dari sensor terdapat *noise* 0,5 Volt saja maka selisih dari nilai yang sesungguhnya (nilai tanpa *noise*) akan semakin besar sesuai resolusinya. Berdasarkan Tabel 5.1, peneliti menentukan untuk memilih ADC 10bit karena memiliki rentang yang jauh lebih besar dari ADC 8bit sehingga lebih presisi, dan juga jika terdapat *noise* 0,1 Volt maka selisih nilai sesungguhnya dengan nilai ADC 10bit hanya mencapai 20 nilai, sedangkan jika menggunakan ADC 12bit akan memiliki selisih mencapai 81 nilai. Selain itu, pemilihan ADC 10bit ini juga

dilakukan pada penelitian sebelumnya yang dilakukan oleh Nuraini N.U. (2017). Oleh karena itu, pada penelitian ini akan menggunakan ADC 10bit yang sudah tertanam pada Arduino Uno.

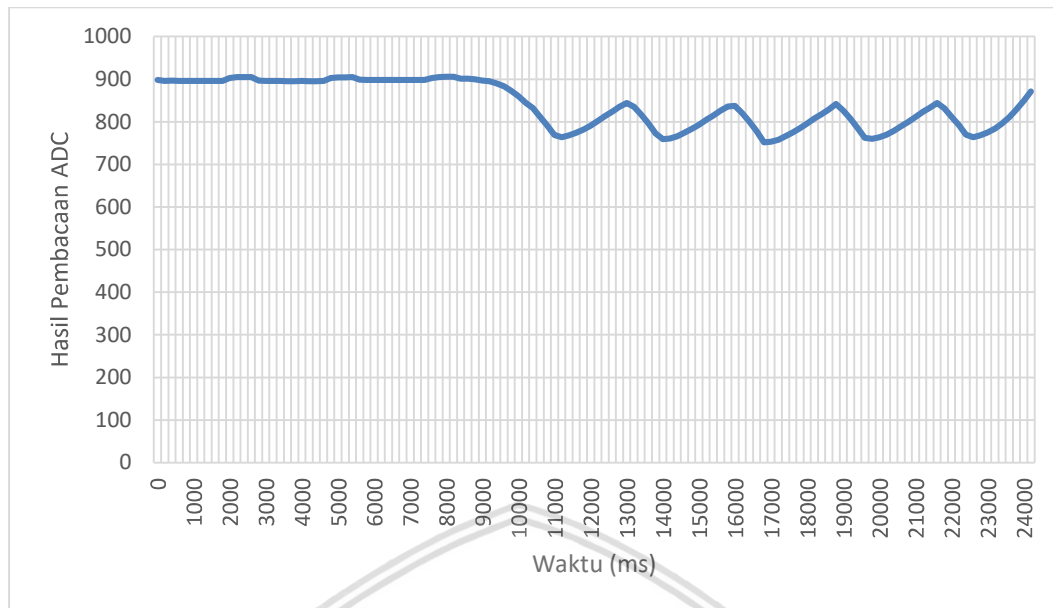
Berdasarkan penelitian Nuraini (2017) sensor *proximity* kapasitif dan *proximity* induktif mampu berjalan dengan menggunakan sumber tegangan 5V yang diperoleh dari Arduino UNO sehingga peneliti melakukan sebuah percobaan untuk mengetahui bagaimana fungsionalitas sensor *proximity* pada saat menggunakan tegangan 5V. Pertama, percobaan dilakukan pada sensor *proximity* kapasitif. Sensor *proximity* yang digunakan memiliki sekrup kecil dan sebuah lampu indikator yang berada di belakang, sekrup tersebut dapat digunakan untuk mengatur tingkat sensitif pada sensor. Saat sensor terhubung dengan sumber tegangan 5V, sekrup pada sensor dapat diputar sehingga sensor *proximity* kapasitif hanya mampu membaca benda yang memiliki kandungan air seperti daun, buah, atau air dalam botol. Keberhasilan tersebut dapat diketahui dengan memperhatikan lampu indikator, dimana saat benda yang mengandung air didekatkan pada permukaan sensor maka lampu indikator akan menyala, dan selain itu lampu indikator tidak akan menyala. Dari konfigurasi yang dilakukan tersebut, sensor mampu membaca benda dengan yang memiliki kandungan air dengan baik pada tegangan 5V. Hasil percobaan tersebut akan ditunjukkan pada bab pengujian poin 6.3.7. Berdasarkan percobaan tersebut, maka sensor *proximity* kapasitif dapat langsung dihubungkan dengan Arduino UNO dengan sambungan seperti pada Tabel 5.2.

Tabel 5.2 Konfigurasi pin sensor *proximity* kapasitif

Pin <i>Proximity</i> Kapasitif	Pin Arduino UNO
VCC (cokelat)	5V
GND (biru)	GND
VOOUT (hitam)	A1

Tabel 5.2 menunjukkan konfigurasi pin sensor *proximity* kapasitif, dimana pada sistem ini menggunakan sensor *proximity* kapasitif CR18-8DP yang mempunyai 3 buah kabel, yaitu VCC dengan kabel berwarna cokelat, GND dengan kabel biru dan VOOUT dengan kabel hitam. Berdasarkan percobaan sebelumnya, karena sensor *proximity* induktif dapat bekerja dengan baik pada tegangan 5V sehingga pin VCC pada *proximity* kapasitif dihubungkan dengan pin 5V Arduino UNO, GND dari sensor dihubungkan dengan pin GND Arduino, dan pin VOOUT dari sensor akan dihubungkan dengan pin A1 pada Arduino UNO.

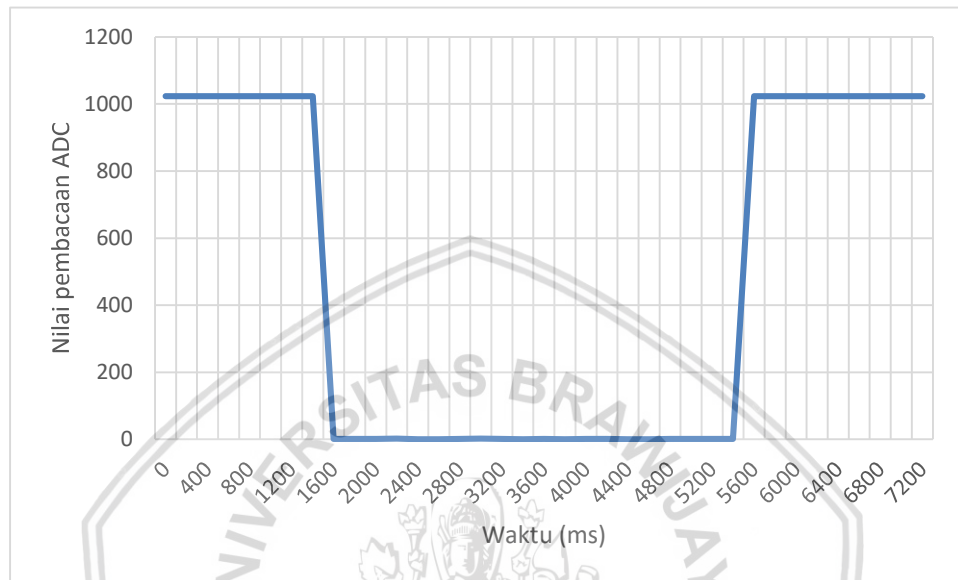
Kemudian percobaan dilakukan terhadap sensor *proximity* induktif dengan sumber tegangan 5V. Sensor *proximity* yang digunakan tidak memiliki sekrup seperti pada sensor *proximity* kapasitif, sehingga tidak ada konfigurasi awal yang dibutuhkan. Hasil pembacaan tersebut ditunjukkan pada Gambar 5.3 dimana data dibaca setiap 200ms sekali.



Gambar 5.3 Percobaan fungsional sensor *proximity* induktif dengan tegangan 5V

Sebuah benda logam didekatkan pada sensor *proximity* induktif pada milisekon ke-8500 hingga percobaan selesai. Pada saat sensor tersebut mendeteksi adanya sebuah benda logam, sensor menghasilkan tegangan keluaran yang tidak stabil. Berdasarkan percobaan tersebut, hipotesis yang dapat diperoleh yaitu sensor tidak dapat bekerja secara maksimal karena kurangnya tegangan yang dibutuhkan, yaitu 6V. Modul *step up* XL6009E merupakan sebuah modul yang mampu meningkatkan tegangan sumber menjadi lebih tinggi sesuai dengan kebutuhan. Tegangan keluaran dari XL6009E dapat disesuaikan dengan cara memutar sekrup kecil pada modul tersebut. Perlu digunakan alat bantu seperti multimeter digital saat memutar sekrup hingga menghasilkan tegangan yang diinginkan. Percobaan kembali dilakukan dengan menguji sensor *proximity* induktif menggunakan tegangan masukan sebesar 6V yang diperoleh dari modul XL6009E. Hasil dari percobaan tersebut ditunjukkan pada Gambar 5.4. Berdasarkan percobaan tersebut, sensor mampu membaca dan menghasilkan tegangan keluaran yang stabil. Pada saat benda yang mengandung logam didekatkan pada area pembacaan sensor (pada milisekon ke-1600 hingga 5400), nilai dihasilkan menjadi 0 dan pada saat sensor tidak mendeteksi adanya kandungan logam, sensor akan menghasilkan nilai 1023. Pada umumnya sensor akan menghasilkan tegangan keluaran maksimal senilai dengan tegangan masukan yaitu 6V, sehingga dilakukan sebuah pengukuran menggunakan multimeter pada tegangan keluaran dari sensor *proximity* induktif, menunjukkan bahwa tegangan keluaran sensor tersebut mencapai sekitar 5,23V. Karena tegangan maksimal yang mampu dibaca oleh ADC pada Arduino UNO yaitu 5V, maka dapat dilakukan penyesuaian terhadap tegangan keluaran tersebut dengan mengatur kembali masukan untuk sensor *proximity* induktif hingga menghasilkan

tegangan maksimal 5V. Setelah tegangan keluaran disesuaikan, hasil pengukuran tegangan masukan untuk sensor tersebut yaitu 5,63V. Kemudian dilakukan percobaan mengenai fungsional pembacaan benda logam pada sensor *proximity* induktif seperti percobaan sebelumnya. Hasil percobaan menghasilkan data yang serupa dengan percobaan sebelumnya, sehingga dapat disimpulkan bahwa sensor *proximity* induktif dapat bekerja dengan baik pada tegangan 5,63V.



Gambar 5.4 Percobaan fungsional sensor *proximity* induktif dengan tegangan 6V

Berdasarkan percobaan tersebut, sensor *proximity* induktif membutuhkan sumber tegangan dari modul XL6009E dan tegangan keluaran dari sensor langsung dihubungkan dengan pin analog Arduino UNO. Sambungan pin modul *step up* XL6009E dengan Arduino UNO dirancang berdasarkan Tabel 5.3 dengan diatur tegangan keluarannya menjadi 5,63V. Sedangkan sambungan Untuk konfigurasi sensor *proximity* kapasitif dan *proximity* induktif dapat dilihat pada Tabel 5.4.

Tabel 5.3 Konfigurasi pin modul *step up* XL6009E dengan Arduino UNO

Pin modul <i>step up</i> XL6009E	Pin Arduino UNO
VIN+	5V
VIN-	GND

Tabel 5.4 Konfigurasi pin sensor *proximity* induktif

Pin <i>Proximity</i> Induktif	Pin perangkat lain
VCC (cokelat)	VOUT+ (Modul XL6009E)
GND (biru)	VOUT- (Modul XL6009E)
VOUT (hitam)	A2 (Arduino UNO)

Tabel 5.4 menunjukkan konfigurasi pin sensor *proximity* induktif Lj12a3-4-Z yang mempunyai 3 buah pin yaitu VCC, GND dan VOUT. Seperti sensor *proximity* kapasitif, kebutuhan tegangan sensor *proximity* induktif juga disesuaikan berdasarkan *datasheet* dari sensor, pin VCC dan GND pada *proximity* induktif juga dihubungkan dengan VOUT+ dan VOUT- dari modul XL6009E. Hasil pembacaan dari sensor (VOUT) dari sensor dihubungkan langsung dengan Arduino UNO pada pin A2.

Berdasarkan percobaan yang telah dilakukan pada kedua sensor *proximity*, maka dapat disimpulkan bahwa keluaran dari kedua sensor tersebut memiliki keluaran maksimum sebesar 5V dan keluaran minimum sebesar 0V. Dengan menggunakan ADC yang sama dengan ADC 10bit pada sensor LDR, maka dapat diketahui hasil pembacaan ADC maksimum berdasarkan perhitungan di bawah, dimana resolusi dari ADC yang digunakan menggunakan hasil dari perhitungan sebelumnya.

$$\text{nilai } ADC_{maks} = \frac{V_{outPS}}{\text{Resolusi AD}} = \frac{5 \text{ V}}{0,004883 \text{ V}} = 1023$$

Sedangkan untuk mengetahui nilai ADC minimum dari sensor *proximity* dapat diketahui berdasarkan perhitungan di bawah ini.

$$\text{nilai } ADC_{min} = \frac{V_{outPS}}{\text{Resolusi AD}} = \frac{0 \text{ V}}{0,004883 \text{ V}} = 0$$

Berdasarkan perhitungan tersebut, maka hasil konversi yang dilakukan oleh ADC terhadap keluaran sensor *proximity* kapasitif dan induktif memiliki rentang nilai 0 hingga 1023.

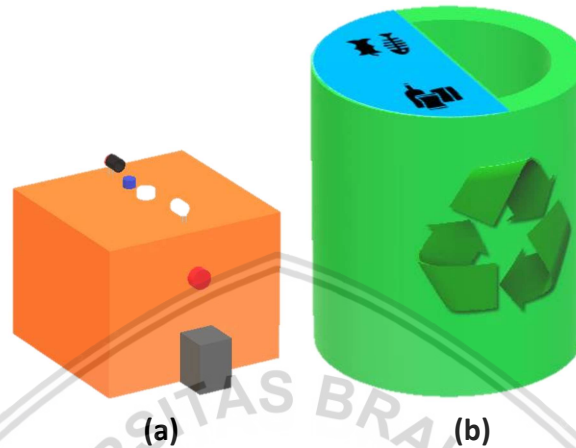
Salah satu kaki *switch* yang digunakan dihubungkan dengan 5V Arduino dan kaki lainnya dihubungkan pin digital Arduino UNO dan kaki positif dari lampu LED, sehingga saat *switch* dalam kondisi menghubungkan, pin Arduino akan membaca data tersebut dengan nilai 1 atau senilai dengan konstanta variabel “HIGH” serta LED akan menyala, dan saat kondisi terputus Arduino akan membaca masukan tersebut sebagai nilai 0 atau senilai dengan konstanta variabel “LOW” dan LED akan mati. Kaki negatif dari LED dihubungkan dengan pin GND Arduino UNO.

Servo yang digunakan pada sistem dihubungkan dengan Arduino UNO berdasarkan Tabel 5.5.

Tabel 5.5 Konfigurasi servo terhadap Arduino UNO

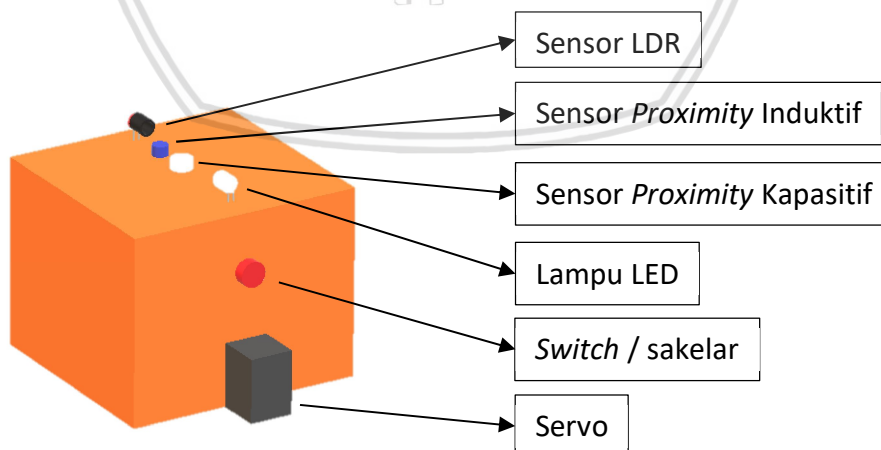
Servo	Pin Arduino
Control (jingga)	9
VCC (merah)	5V
GND (cokelat)	GND

Servo hanya dapat digerakkan dengan PWM dari beberapa pin Arduino UNO yang mendukung keluaran berupa sinyal PWM. Arduino UNO memiliki 6 pin yang mendukung PWM yaitu pin 3, 5, 6, 9, 10, dan 11. Pada penelitian ini pin *control* pada servo dihubungkan ke pin 9. Sedangkan untuk VCC dihubungkan dengan pin 5V Arduino UNO dan GND servo dihubungkan dengan pin GND Arduino UNO.

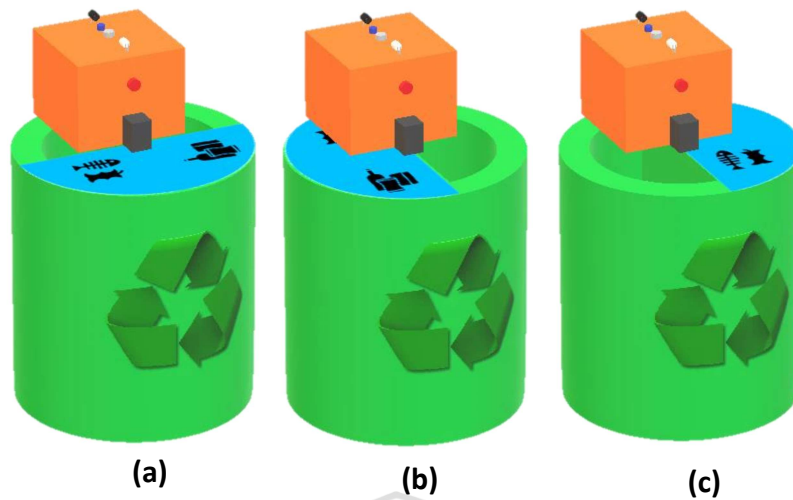


Gambar 5.5 (a) Desain sistem dan (b) desain tempat sampah

Setelah selesai tahap perancangan sistem, kemudian membuat desain *prototype* alat. Gambar 5.5 (a) merupakan desain sistem dari alat yang mengandung komponen seperti yang tertera pada skema Gambar 5.2, alat ini didesain berbentuk kubus dengan posisi semua sensor berada pada bagian tengah atas dari kubus tersebut, dan sebuah servo yang berada di depan kubus tersebut memegang sebuah tutup tempat sampah yang berbentuk setengah lingkaran. Sedangkan pada Gambar 5.5 (b) menggambarkan tempat sampah sederhana yang memiliki sekat di tengahnya sebagai pemisah ruang untuk sampah organik dan anorganik. Untuk desain peletakan sensor dijelaskan pada Gambar 5.6.

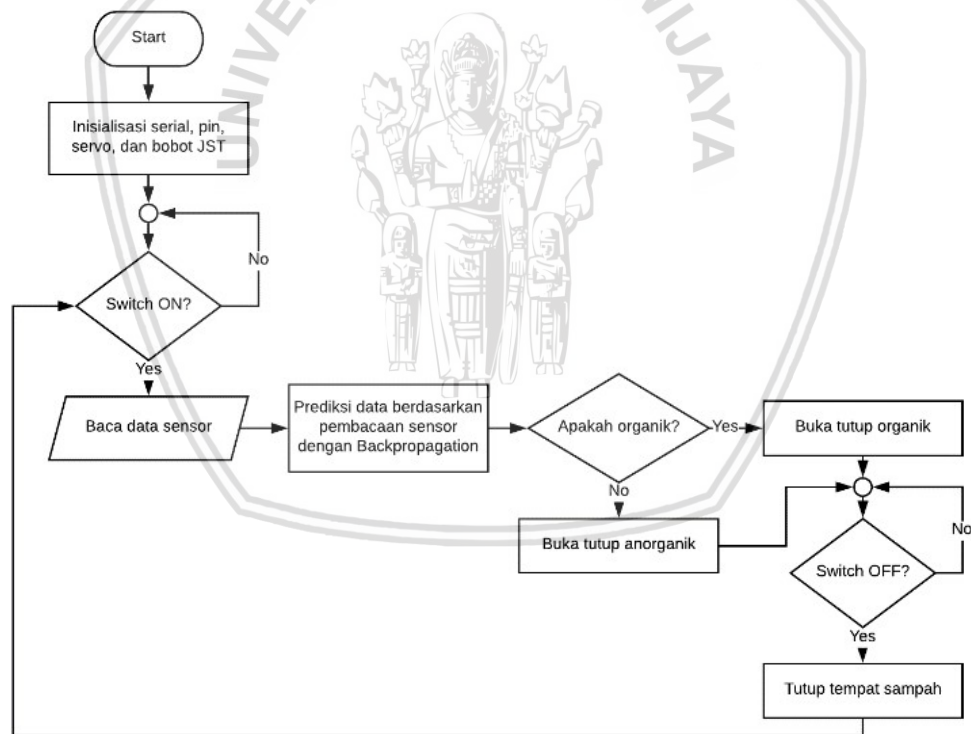


Gambar 5.6 Ilustrasi desain sistem



Gambar 5.7 (a) Kondisi saat tempat sampah tertutup, (b) kondisi saat sistem buka tutup dari ruang anorganik, (c) kondisi saat sistem membuka tutup dari ruang sampah organik

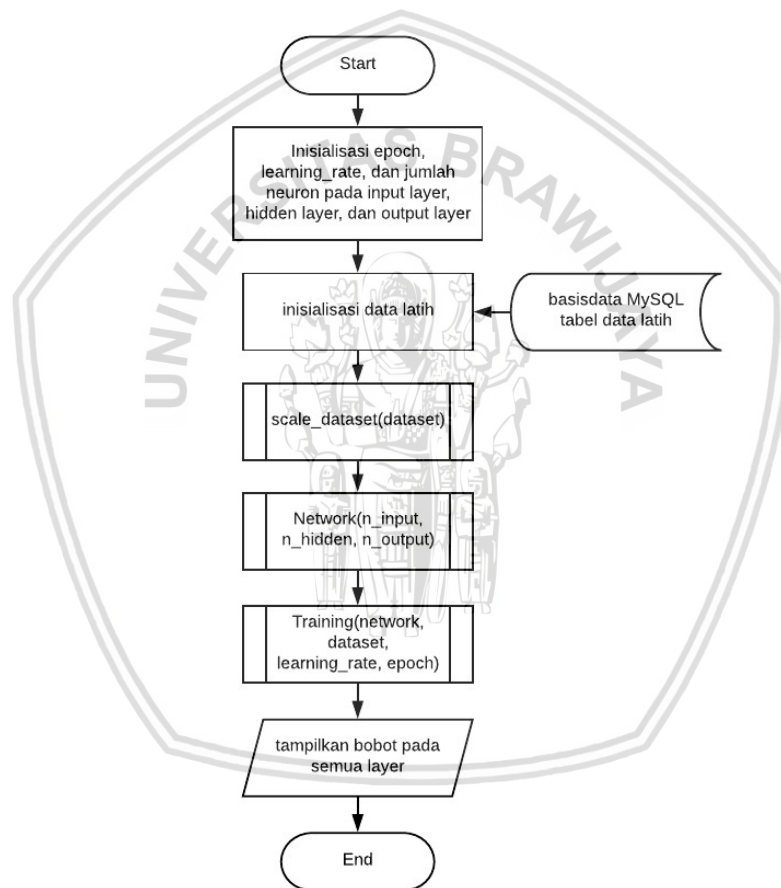
5.1.2 Perancangan Perangkat Lunak



Gambar 5.8 Diagram alir sistem

Pada sub bab ini, menjelaskan tentang perancangan sistem dari segi perangkat lunak yang meliputi penjelasan dari diagram alir program yang digunakan. Pada sistem ini, program dibuat menggunakan PlatformIO IDE dan PyCharm. Diagram alir pada sistem utama dijelaskan pada Gambar 5.8. Diagram alir pada tersebut menjelaskan alur kerja dari program pada Arduino. Saat

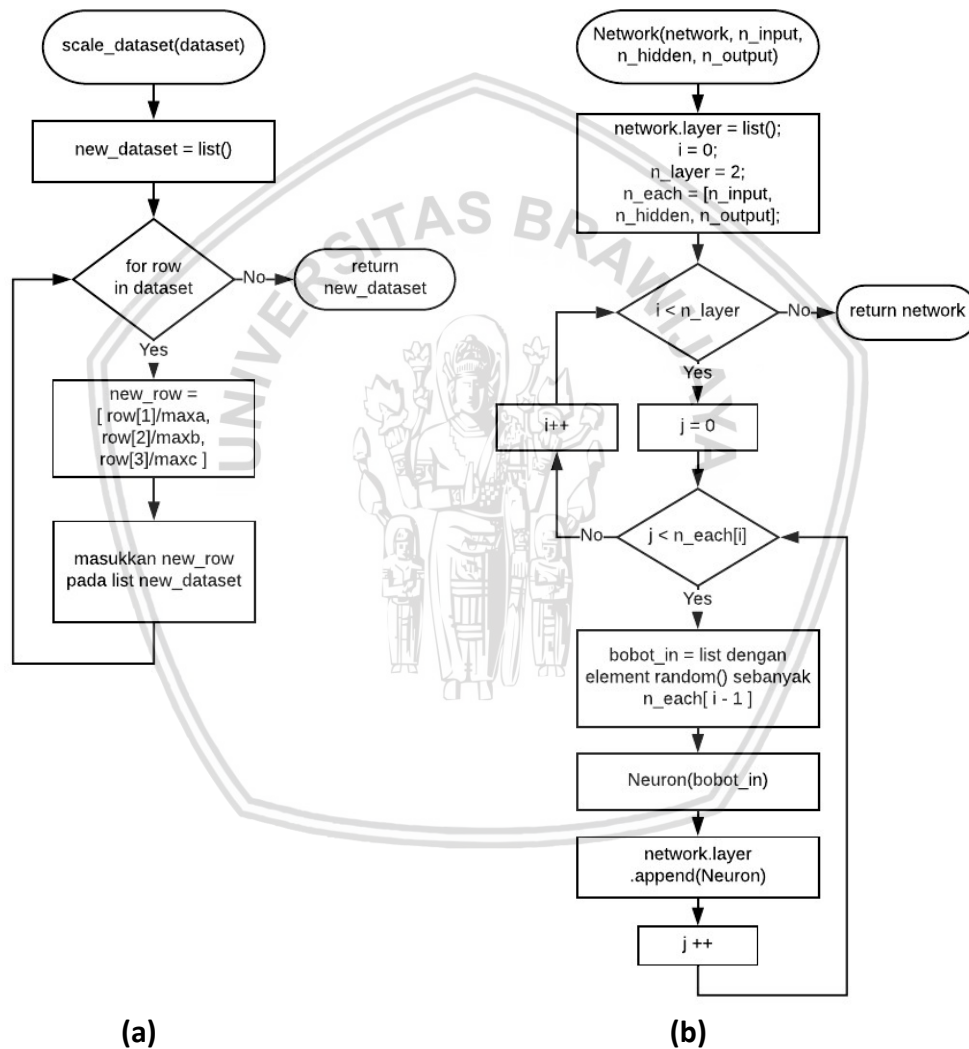
program dimulai, program akan melakukan inisialisasi serial, pin, servo, serta bobot yang didapat dari hasil inisialisasi. Kemudian sistem akan menunggu hingga *switch* disambungkan. Jika *switch* tersambung, sistem akan mengakuisisi data dari sensor, dan melakukan prediksi dengan metode jaringan saraf tiruan *backpropagation* berdasarkan bobot yang diperoleh dari hasil pelatihan. Prediksi dilakukan dengan menggunakan rumus *feed forward* fase 1 pada langkah *backpropagation*. Jika sistem memprediksi sampah organik, maka program akan memutar piringan servo hingga 90 derajat berlawanan arah jarum jam. Namun jika sistem memprediksi anorganik, maka program akan memutar piringan servo hingga 90 derajat searah arah jarum jam. Setelah membuka tutupnya, sistem akan kembali membaca *switch* hingga dinonaktifkan oleh pengguna dan sistem akan menutup tempat sampah.



Gambar 5.9 Diagram alir pelatihan *backpropagation*

Diagram alir pada Gambar 5.9 menjelaskan alur program yang diterapkan pada komputer yang terpasang python 3.6 dan PyCharm IDE. Program ini tidak diterapkan pada sistem alat, namun berjalan terpisah pada perangkat komputer. Program ini dibutuhkan untuk melakukan pelatihan data dan memperoleh bobot dari hasil pelatihan. Ketika program dimulai, program akan melakukan inisialisasi *epoch*, *learning rate* serta jumlah neuron pada lapis masukan, lapis tersembunyi

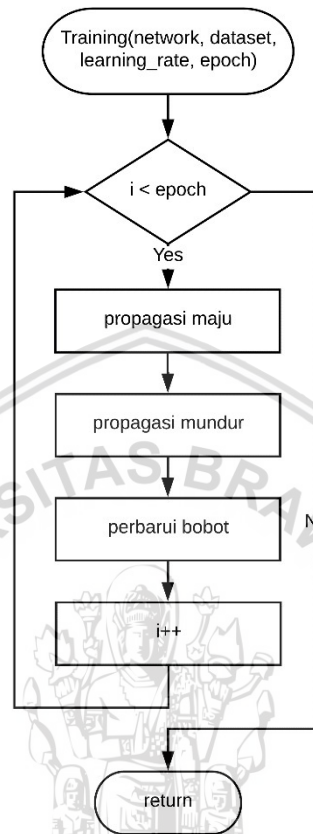
dan lapis keluaran, serta sistem akan mengambil data latih yang tersimpan pada basis data MySQL. Setelah data latih diambil, data latih tersebut kemudian diskalakan hingga memiliki batas terkecil sebesar 0 dan batas terbesar yakni 1 dengan memanggil fungsi `scale_dataset`. Setelah diskalakan, program akan membangun arsitektur jaringan saraf tiruan serta melakukan inisialisasi bobot dan bias secara acak dengan memanggil fungsi `Network`. Setelah arsitektur dibangun, program akan melakukan pelatihan *backpropagation* dengan memanggil fungsi `Training`. Setelah fungsi `Training` selesai dijalankan, bobot hasil pelatihan akan ditampilkan pada jendela terminal.



Gambar 5.10 (a) diagram alir fungsi `scale_dataset`, dan (b) diagram alir fungsi `Network`

Gambar 5.10 merupakan isi dari fungsi-fungsi yang dijelaskan pada Gambar 5.9. Gambar 5.10 (a) merupakan diagram alir untuk mengskalakan setiap data yang ada dengan cara data asli dibagi dengan maksimal data pada data. Gambar 5.9 (b) merupakan diagram alir dalam membangun arsitektur jaringan

saraf tiruan. Pada diagram alir tersebut, satu lapisan merupakan *array* yang menyimpan data setiap neuronnya dimana setiap neuron memuat nilai bobot awal secara acak sebanyak jumlah data yang masuk dari lapisan sebelumnya, dan variabel `network.layer` menyimpan semua lapisan dalam bentuk *array*.



Gambar 5.11 Diagram alir fungsi Training

Gambar 5.11 merupakan diagram alir dari fungsi Training yang dipanggil oleh program utama yang digambarkan pada Gambar 5.8 setelah memanggil fungsi Network. Fungsi tersebut bertujuan untuk melakukan pelatihan *backpropagation* fase 1 (propagasi maju) hingga fase 3 (perbarui bobot).

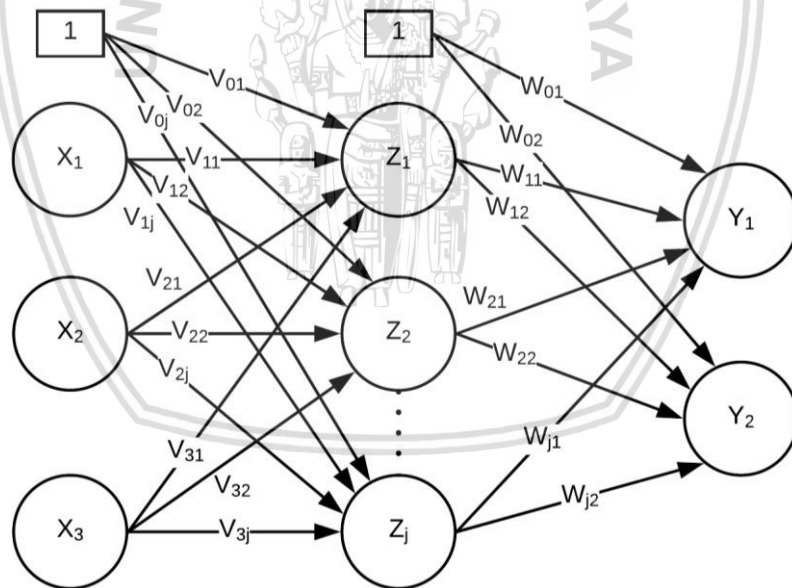
5.1.3 Perancangan Jaringan Saraf Tiruan

Tahap perancangan jaringan saraf tiruan ini bertujuan untuk menentukan bagaimana arsitektur jaringan saraf tiruan yang digunakan, bagaimana pengumpulan data latih, serta bagaimana menentukan *epoch* dan *learning rate* yang dibutuhkan oleh sistem.

5.1.3.1 Arsitektur Jaringan Saraf Tiruan

Pada penelitian ini memiliki 3 buah parameter yang dapat digunakan untuk dapat melakukan klasifikasi berdasarkan sifat kedua jenis sampah tersebut, di antaranya kandungan air di dalamnya, tingkat transparansi, serta kandungan logam. Oleh karena sistem ini memiliki 3 parameter masukan, maka pada lapis

masukannya menggunakan 3 unit neuron yang disimbolkan dengan huruf X. Pada penelitian ini, sistem diharapkan mampu membedakan 2 jenis sampah, yakni sampah organik dan sampah anorganik. Maka dapat disimpulkan bahwa hanya ada 2 jenis kelas pada fitur jenis sampah, sehingga membutuhkan 2 unit neuron pada lapis keluaran dimana setiap neuron merepresentasikan jenis sampah organik dan anorganik yang disimbolkan dengan huruf Y. Sedangkan untuk menentukan jumlah lapisan tersembunyi yang disimbolkan dengan huruf Z, peneliti menggunakan arsitektur sederhana yang hanya menggunakan 1 lapis tersembunyi dimana hal tersebut dipilih berdasarkan kinerja dan jumlah memori yang dibutuhkan pada saat prediksi yang dilakukan pada perangkat Arduino dimana semakin banyak jumlah lapisan dan/atau jumlah unit neuron maka akan membutuhkan memori yang semakin besar dan jumlah *clock* yang dibutuhkan untuk menyelesaikan prediksi juga semakin banyak, mengingat bahwa Arduino memiliki keterbatasan pada banyaknya *clock* dan memori yang diperlukan. Untuk mendapatkan akurasi yang lebih maksimal maka akan terdapat perubahan yang dilakukan pada saat pelatihan data, yaitu nilai *epoch*, *learning rate* (tingkat pembelajaran), serta jumlah neuron dalam lapisan tersembunyi untuk mendapatkan tingkat kesalahan terkecil. Percobaan tersebut akan dijelaskan pada poin 5.2.2.2 setelah implementasi perangkat lunak dilakukan. Arsitektur jaringan saraf tiruan tersebut dijelaskan pada Gambar 5.12.



Gambar 5.12 Rancangan jaringan saraf tiruan

5.1.3.2 Pengumpulan Data Latih

Peneliti mengumpulkan beberapa benda organik dan anorganik secara acak. Data setiap benda akan dideteksi oleh sensor LDR, *proximity* induktif, dan *proximity* kapasitif dicatat dan disimpan ke basis data MySQL. Akibat pembacaan sensor yang tidak stabil akibat *noise* yang terjadi, maka peneliti memutuskan untuk mencatat 4 baris data untuk setiap sampah yang terdeteksi oleh sampah

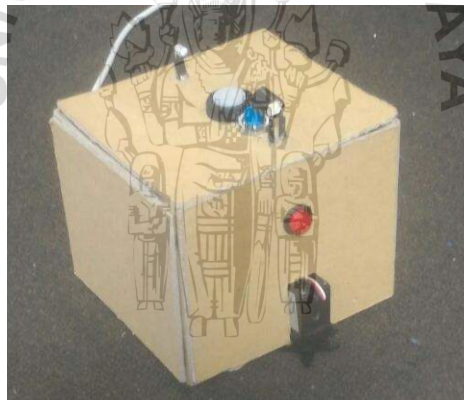
serta dipastikan tidak ada satu baris yang kembar pada basis data. Setiap barisnya merupakan hasil pembacaan sensor secara bersamaan yang tampil pada *serial monitor* dengan *delay* 2 detik untuk setiap barisnya. Pada penelitian ini, peneliti berhasil mengumpulkan 104 data dengan perbandingan 50:50.

5.1.3.3 Pelatihan Jaringan Saraf Tiruan

Hasil akurasi dari prediksi yang dilakukan oleh jaringan saraf tiruan tidak hanya dipengaruhi oleh arsitektur dan data latih, namun penentuan *epoch* dan juga *learning rate* (laju pembelajaran) yang diterapkan pada saat pelatihan data. Nilai *epoch* dan *learning rate* akan diperoleh setelah melakukan beberapa percobaan untuk mendapatkan tingkat kesalahan terkecil dari bobot yang dicari pada poin 5.2.2.2.

5.2 Implementasi Prototipe Alat

Dalam mengimplementasikan prototipe alat pemilah sampah ini mengacu pada perancangan di subbab 5.1.1. Prototipe ini berbentuk balok dengan ukuran $14 \times 13 \times 12 \text{ cm}^3$ dengan bahan karton dengan ketebalan 2,5 mm. Hasil implementasi prototipe beserta peletakan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.6.

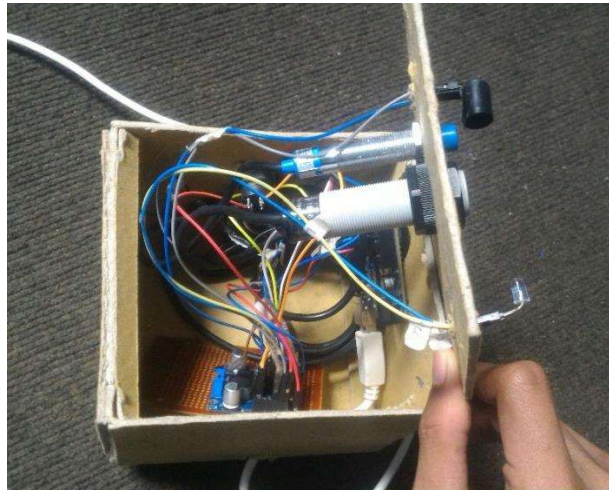


Gambar 5.13 Implementasi desain sistem

Gambar 5.13 menunjukkan prototipe desain dari sistem. Terlihat ada 2 buah sensor yang mengarah ke atas, yaitu sensor *proximity* kapasitif dan sensor kapasitif induktif. Serta ada sebuah sensor LDR yang disinari oleh lampu dari sisi seberangnya.

5.2.1 Implementasi Perangkat Keras

Pada sub bab ini menjelaskan proses dari implementasi perangkat keras yang mencakup komponen elektronik. Sistem pada alat ini memiliki beberapa komponen di dalamnya, yaitu Arduino Uno, servo, LDR, LED, sensor *proximity* kapasitif, dan sensor *proximity* induktif yang disusun berdasarkan skema yang ditunjukkan Gambar 5.2. Implementasi dari perancangan tersebut dapat dilihat pada Gambar 5.14.



Gambar 5.14 Implementasi sistem

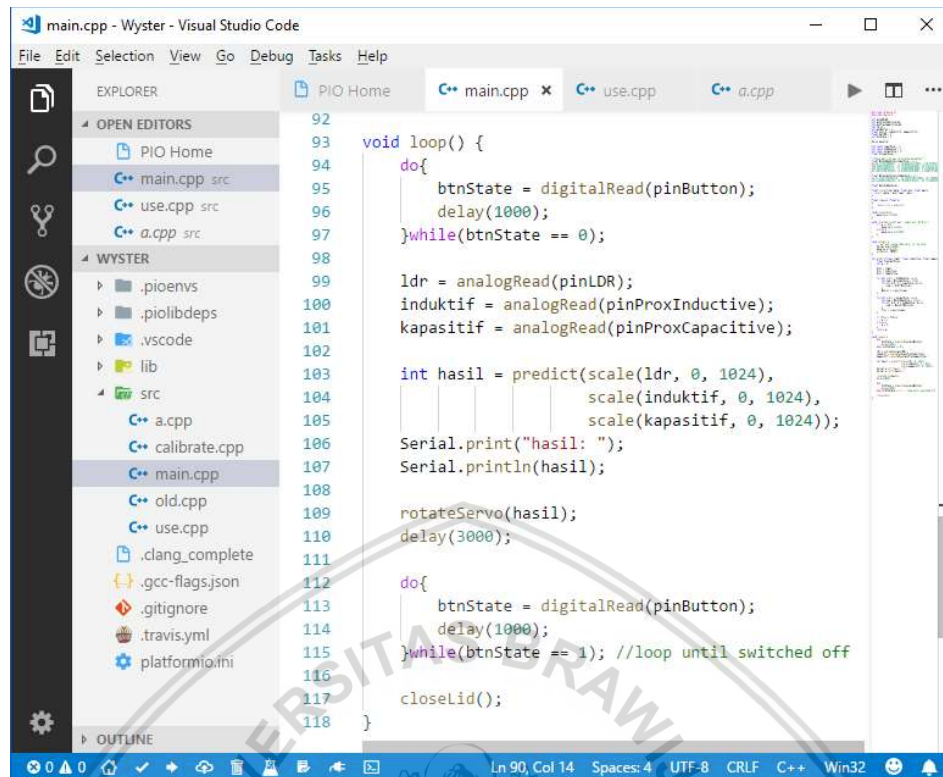
5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada subbab ini menjelaskan proses dalam menjalankan program. Dalam mengimplementasikan perangkat lunak ini, terdapat 2 software yang dipakai yaitu PlatformIO IDE dan PyCharm.

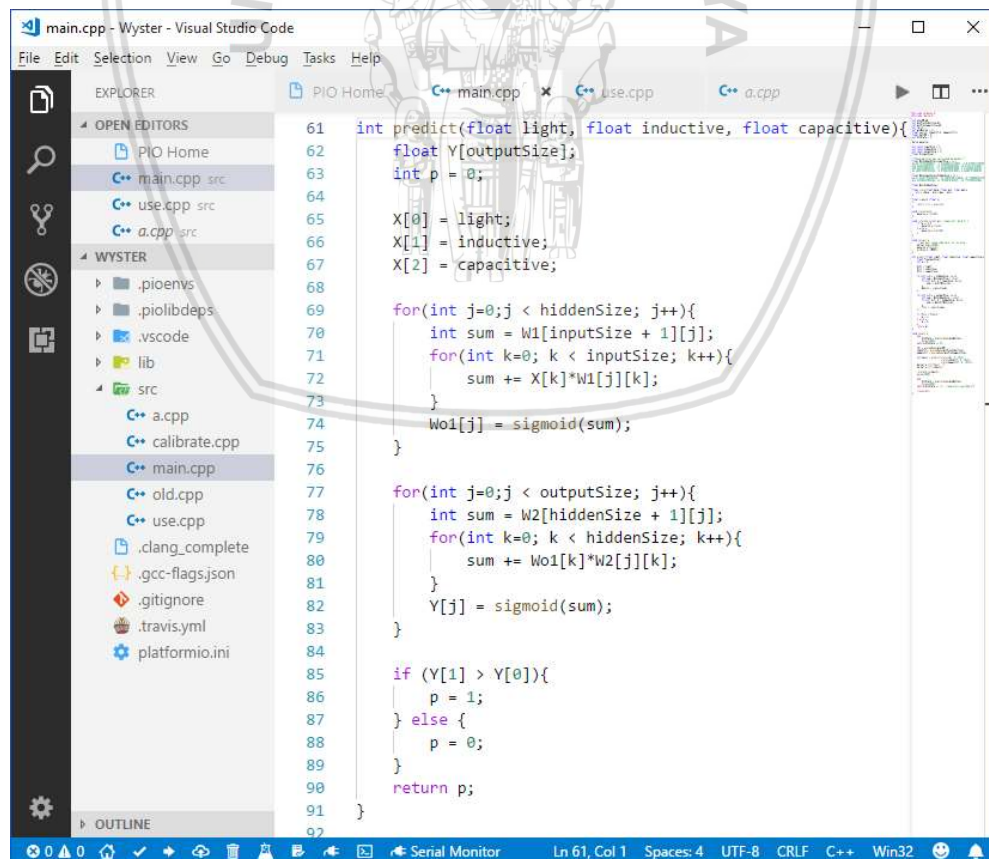
5.2.2.1 Implementasi Perangkat Lunak Pada Mikrokontroler Sistem

Pada implementasi ini, sistem menggunakan PlatformIO IDE untuk memprogram dari mikrokontroler Arduino Uno dengan menggunakan bahasa pemrograman C++. Program Arduino ini dibangun berdasarkan diagram alir yang sebelumnya tunjukkan oleh Gambar 5.7. Potongan program tersebut dapat dilihat pada Gambar 5.15, sedangkan untuk kode program secara keseluruhan dapat dilihat pada lampiran B.1.

Pada Gambar 5.16 dapat terlihat algoritme prediksi dimana menggunakan algoritme propagasi maju (fase 1). Baris 70-73 menerapkan Persamaan (2.1), kemudian menghitung nilai aktivasinya dengan fungsi sigmoid seperti Persamaan (2.2) yang dijalankan oleh baris 74. Hasil dari fungsi aktivasi tersebut menjadi nilai keluaran dari setiap unit neuron dan diulang sebanyak jumlah neuron pada lapis tersembunyi. Langkah selanjutnya yakni menghitung hasil prediksi pada lapis keluaran dengan menggunakan Persamaan (2.3) yang diimplementasikan pada baris 77-83 dan Persamaan (2.4) yang diimplementasikan pada baris 82. Setelah itu akan didapat nilai keluaran yang masih berupa *array* berisi 2 elemen sesuai jumlah *output* yang digunakan. Hasil dari prediksi dapat diketahui dengan mencari nilai terbesar dengan membandingkan elemen pada indeks ke-0 dan ke-1 pada *array* tersebut karena setiap elemen mewakili satu jenis sampah, hal ini diterapkan pada baris 85-89. Setelah hasil prediksi didapatkan, maka nilai prediksi yang disimpan pada variabel *p* akan dikembalikan ke pemanggil fungsi dengan perintah *return* pada baris 90.



Gambar 5.15 Potongan program Arduino

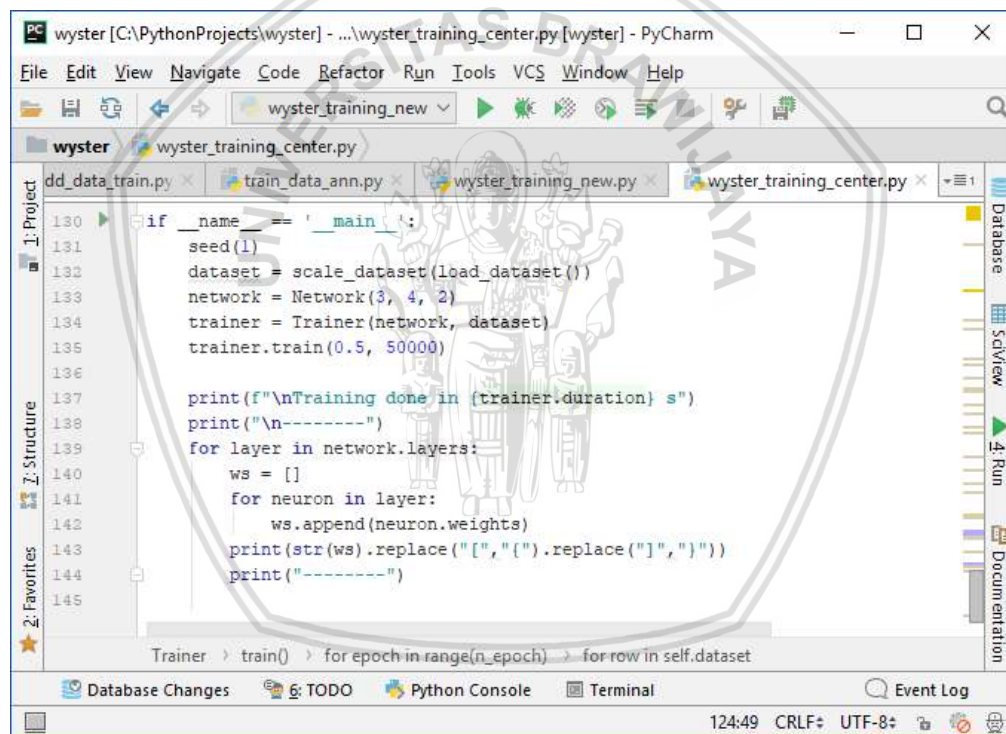


Gambar 5.16 Potongan program prediksi

5.2.2.2 Implementasi Perangkat Lunak Untuk Pelatihan Data

Program untuk pelatihan hanya diimplementasikan pada sistem tambahan berupa perangkat komputer, tahap ini dibutuhkan untuk mencari bobot yang diperlukan untuk melakukan prediksi. Program diimplementasikan pada sistem komputer yang sudah terpasang python 3.6 dan ditulis pada aplikasi PyCharm. Kode program secara lengkap dapat dilihat pada lampiran poin B.2.

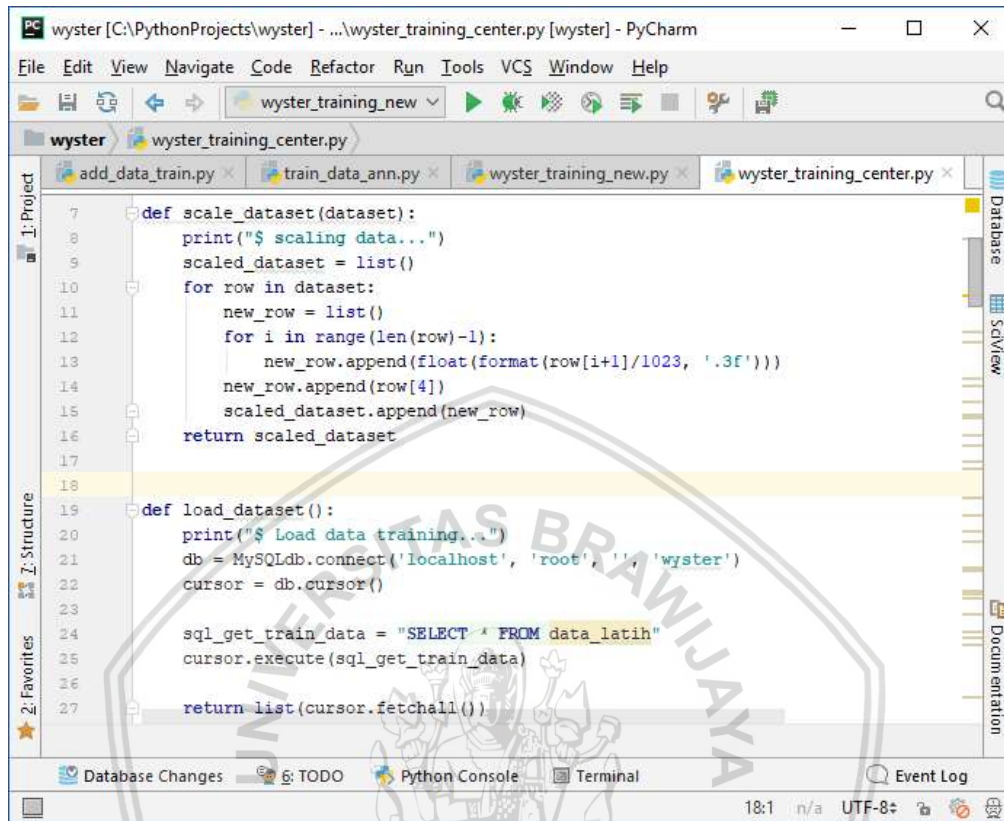
Pada Gambar 5.15 terlihat potongan program yang merupakan fungsi utama dari program (baris 130-144). Baris 132 akan mengambil data latih dari basis data dengan memanggil fungsi `load_dataset` setelah itu melakukan skala data latih sehingga memiliki rentang dari 0 hingga 1 dengan memanggil fungsi `scale_dataset`. Baris 133 bertujuan untuk membangun arsitektur jaringan saraf tiruan dengan melempar parameter jumlah neuron pada lapis masukan, lapis tersembunyi, serta lapis keluaran. Pada baris 134 dilakukan pelatihan dengan melempar *learning rate* dan *epoch*.



Gambar 5.17 Potongan program pada fungsi utama

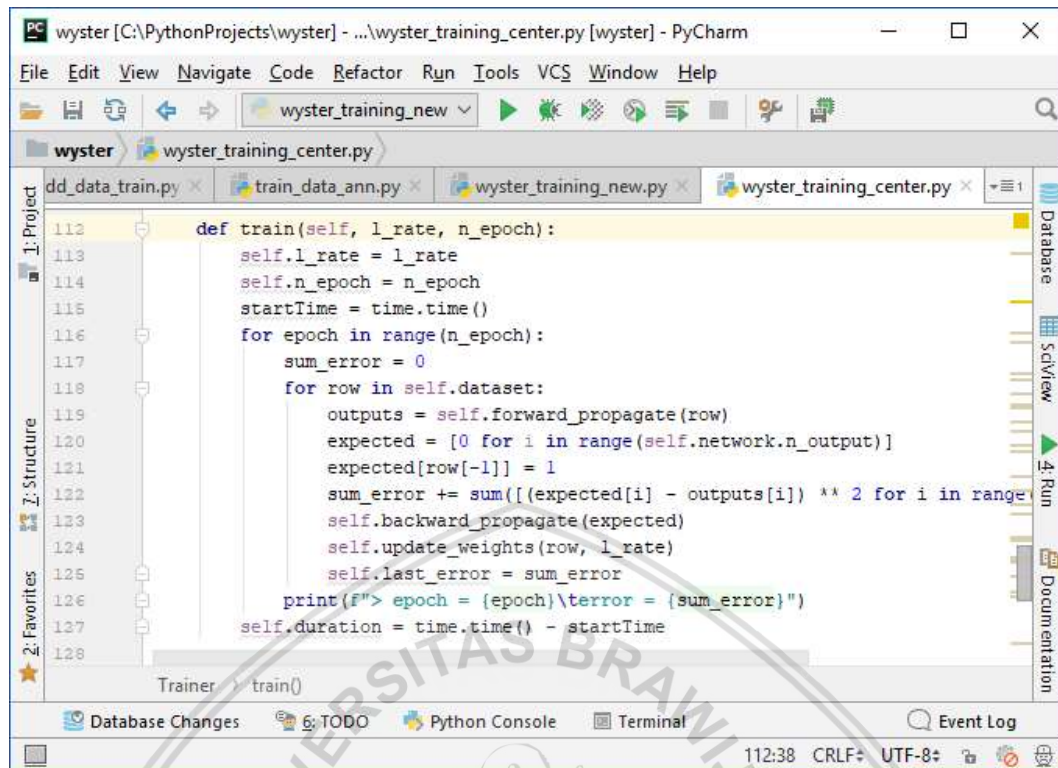
Gambar 5.18 menunjukkan potongan program pada fungsi `scale_dataset` dan fungsi `load_dataset`. Fungsi `load_dataset` (baris 19-27) melakukan konektivitas ke basis data (baris 21-22) dan mengambil seluruh data latih pada basis data dari Tabel bernama `data_latih` yang dijalankan oleh kode pada baris 24-25, kemudian melempar kembalian berupa *list* (baris 27). Sedangkan fungsi `scale_dataset` akan membaca nilai pada setiap baris dan setiap kolom selain kolom label dan

membaginya dengan nilai 1024 (nilai pembacaan maksimal dari ADC Arduino) dan disimpan pada *list* baru.



Gambar 5.18 Potongan program fungsi `scale_dataset` dan fungsi `load_dataset`

Gambar 5.19 menunjukkan potongan program fungsi pelatihan dimana algoritme *backpropagation* benar-benar dijalankan. Pada *backpropagation* terdapat fase propagasi maju (fase 1) dimana pada fase ini semua bobot yang tersimpan akan diuji melewati semua neuron dan mendapatkan hasil keluarannya. Fase tersebut dijalankan dengan memanggil fungsi `forward_propagate` (baris 119), dan kemudian fase propagasi mundur (fase 2) dimana pada fase ini akan dihitung tingkat kesalahan dan selisih yang terjadi berdasarkan data latih dengan keluaran dari hasil perhitungan fase propagasi maju sebelumnya. Fase ini dijalankan dengan memanggil fungsi `backward_propagate` pada baris 123. Selain itu terdapat pula fase pembaruan bobot (fase 3) dimana pada fase ini semua bobot pada jaringan akan diperbarui berdasarkan hasil perhitungan dari fase propagasi mundur. Hal ini dijalankan dengan memanggil fungsi `update_weights` yang dipanggil pada baris 124.



Gambar 5.19 Potongan program fungsi pelatihan

Setelah algoritme *backpropagation* diimplementasikan, program masih membutuhkan nilai *learning rate* dan *epoch* yang sesuai untuk mencapai akurasi maksimum. Berdasarkan data latih yang telah dikumpulkan, peneliti melakukan percobaan dengan menggunakan 3 hingga 5 neuron dalam lapisan tersembunyi, nilai *epoch*: 1000, 5000, 10000, dan 50000, serta beberapa nilai *learning rate*: 0,75, 0,5, 0,25, 0,1 dan 0,05. Setelah percobaan tersebut dilakukan, maka akan didapat kombinasi dengan tingkat kesalahan terkecil yang didapat pada saat pelatihan. Perhitungan kesalahan tersebut diperoleh melalui Persamaan (2.12). Hasil percobaan tersebut ditampilkan pada Tabel 5.5, Tabel 5.6, dan Tabel 5.7. Berdasarkan data dari ketiga Tabel tersebut, diketahui bahwa kombinasi 4 neuron pada lapis tersembunyi, dengan *learning rate* 0,5 dan *epoch* 50000 memiliki tingkat kesalahan yang terkecil yaitu 3,266.

Tabel 5.6 Tingkat kesalahan (error) berdasarkan epoch dan learning rate (lr) dengan 3 neuron pada lapisan tersembunyi

epoch \ lr	0.75	0.5	0.25	0.1	0.05
1000	5.388167	5.476851	5.683395	7.259616	10.297236
5000	5.078370	4.954245	5.160383	5.320715	5.379178
10000	4.920218	4.940761	4.675926	4.812052	5.241328
50000	3.428355	3.606788	3.423912	3.350133	3.375624

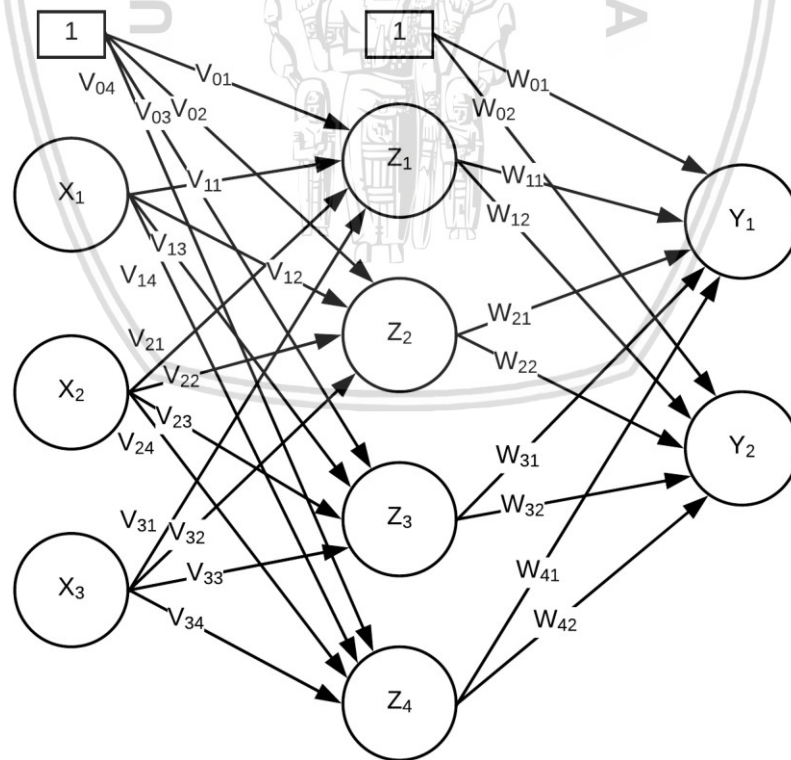
Tabel 5.7 Tingkat kesalahan (*error*) berdasarkan *epoch* dan *learning rate (lr)* dengan 4 neuron pada lapisan tersembunyi

epoch \ lr	0.75	0.5	0.25	0.1	0.05
1000	5.377300	5.445340	5.734783	6.733620	8.616023
5000	5.027765	5.027175	5.141685	5.230545	5.517859
10000	4.813095	4.311294	4.713789	4.929141	5.240145
50000	3.451588	3.265895	3.341168	3.391144	3.432128

Tabel 5.8 Tingkat kesalahan (*error*) berdasarkan *epoch* dan *learning rate (lr)* dengan 5 neuron pada lapisan tersembunyi

epoch \ lr	0.75	0.5	0.25	0.1	0.05
1000	5.403262	5.539196	5.651939	6.969973	8.748111
5000	4.983618	5.005424	5.098925	5.228495	5.466394
10000	4.539722	4.049211	4.203761	5.114719	5.241439
50000	3.453207	3.276931	3.291072	3.384566	3.471869

Berdasarkan percobaan di atas, arsitektur jaringan saraf tiruan yang dibangun dapat dilihat melalui Gambar 5.20.



Gambar 5.20 Arsitektur jaringan saraf tiruan yang digunakan

BAB 6 PENGUJIAN

Bab ini membahas mengenai pengujian pada sistem yang dibuat. Pengujian meliputi pengujian akurasi setiap sensor, pengujian terhadap akurasi klasifikasi yang dilakukan, serta pengujian waktu tempuh pada saat melakukan pelatihan dan saat melakukan prediksi.

6.1 Pengujian Fungsional Setiap Sensor

Subbab ini akan membahas pengujian fungsionalitas pada masing-masing sensor apakah berjalan sesuai harapan atau tidak. Ada 3 sensor yang diuji, yaitu sensor LDR, sensor *proximity* induktif, serta sensor *proximity* kapasitif. Nilai hasil pembacaan sensor-sensor tersebut merupakan hasil konversi perubahan tegangan oleh ADC 10bit yang terdapat pada Arduino Uno, sehingga memiliki rentang nilai 0 hingga 1023.

6.1.1 Sensor LDR (Light Dependent Resistor)

Pada pengujian ini menggunakan sensor LDR yang diterapkan pada sistem untuk mendeteksi tingkat transparansi suatu objek yang terdeteksi. Tingkat transparansi ini dapat dideteksi berdasarkan tingkat cahaya yang mampu melewati benda hingga dapat diterima oleh sensor LDR.

6.1.1.1. Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui fungsionalitas dari sensor LDR dalam mendeteksi tingkat transparansi suatu objek.

6.1.1.2. Prosedur pengujian

Pengujian ini dilakukan secara terurut berdasarkan prosedur berikut:

1. Menghubungkan sistem ke komputer melalui kabel USB
2. Melakukan *upload* program yang dapat mengirim hasil pembacaan sensor secara serial menggunakan PlatformIO IDE ke sistem
3. Membuka *Serial Monitor* pada Platform IDE
4. Mendekatkan sampah ke sensor hingga cahaya dari lampu LED tertutupi
5. Mengamati perubahan nilai dari sensor yang tampil pada *Serial Monitor* PlatformIO IDE

6.1.1.3. Hasil pengujian

Setelah melakukan pengujian sebanyak 16 kali maka diperoleh hasil seperti pada Tabel 6.1.

Tabel 6.1 Hasil Pembacaan sensor LDR

Percobaan ke-	Objek	Transparansi	Hasil Sensor
1	Plastik bening	Ya	787
2	Plastik obat (bening)	Ya	864
3	Gelas kaca	Ya	880
4	Kantong plastik putih	Ya	364
5	Kantong plastik kuning	Ya	218
6	Botol <i>tupperware</i> hitam	Ya	142
7	Selemba kertas HVS	Ya	413
8	Kain lap tangan	Tidak	68
9	Akrilik hitam	Tidak	14
10	Tahu	Tidak	12
11	Timun	Tidak	13
12	Spidol	Tidak	8
13	Pensil	Tidak	14
14	<i>Stapler</i>	Tidak	13
15	Tisu	Ya	197
16	Gunting	Tidak	9

6.1.1.4. Analisis

Berdasarkan Tabel 6.1, terlihat bahwa sensor LDR dapat mendeteksi transparansi benda, pada benda yang tidak transparan maka hasil sensor akan menghasilkan nilai yang kecil, semakin tinggi tingkat transparansi benda tersebut maka akan semakin tinggi juga nilai hasil pembacaan yang dilakukan oleh sensor LDR.

6.1.2 Sensor *Proximity* Induktif

Pada penelitian ini, *proximity* induktif bertindak untuk mendeteksi adanya logam pada benda yang dideteksi. *Proximity* yang digunakan memiliki nilai awal 1023.

6.1.2.1 Tujuan pengujian

Pengujian ini berfungsi untuk mengetahui fungsionalitas sensor *proximity* induktif yang digunakan dengan melihat respons dan akurasi berdasarkan nilai yang dihasilkan.

6.1.2.2 Prosedur pengujian

Pengujian ini dilakukan secara terurut berdasarkan prosedur berikut:

1. Menghubungkan sistem ke komputer melalui kabel USB
2. Melakukan *upload* program yang dapat mengirim hasil pembacaan sensor secara serial menggunakan PlatformIO IDE ke sistem
3. Membuka *Serial Monitor* pada PlatformIO IDE
4. Mendekatkan sampah ke permukaan sensor
5. Mengamati perubahan nilai dari sensor yang tampil pada *serial monitor* PlatformIO IDE

6.1.2.3 Hasil percobaan

Setelah melakukan percobaan sebanyak 10 kali maka didapat hasil seperti yang ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil pembacaan sensor *proximity* induktif

Percobaan ke-	Objek	Mengandung logam	Hasil Sensor
1	Tahu	Tidak	1023
2	Pensil	Tidak	1023
3	Gunting	Ya	2
4	Pita perekat hitam	Tidak	1023
5	Kertas HVS	Tidak	1023
6	<i>Stapler</i>	Ya	0
7	Gunting kuku	Ya	2
8	Spidol papan tulis	Tidak	1023
9	Baterai Kotak	Ya	0
10	Botol <i>tupperware</i>	Tidak	1023

6.1.2.4 Analisis

Pada Tabel 6.2 terlihat bagaimana pola kerja sensor *proximity* induktif. Ketika sensor mendeteksi adanya kandungan logam pada sampah tersebut, maka sensor akan mendeteksinya dengan menghasilkan nilai yang mendekati nilai 0, sedangkan ketika sampah tersebut tidak mengandung logam, maka sensor tidak

dapat mendeteksinya dan menghasilkan nilai berkisar seperti kondisi awal yaitu 1023.

6.1.3 Sensor *Proximity* Kapasitif

Proximity kapasitif berfungsi untuk mendeteksi nilai kapasitans dari sebuah objek. Pada penelitian ini *proximity* kapasitif dapat berfungsi untuk mendeteksi adanya cairan pada sampah yang dideteksi.

6.1.3.1 Tujuan Pengujian

Pengujian ini berfungsi untuk mengetahui fungsionalitas sensor *proximity* kapasitif yang digunakan dengan melihat respons dan akurasi berdasarkan nilai yang dihasilkan oleh sensor.

6.1.3.2 Prosedur pengujian

Untuk pengujian ini dilakukan secara terurut berdasarkan prosedur berikut:

1. Menghubungkan sistem ke komputer melalui kabel USB
2. Melakukan *upload* program yang dapat mengirim hasil pembacaan sensor secara serial menggunakan PlatformIO IDE ke sistem
3. Membuka *Serial Monitor* pada PlatformIO IDE
4. Mendekatkan sampah ke permukaan sensor
5. Mengamati perubahan nilai dari sensor yang tampil pada *Serial Monitor* PlatformIO IDE

6.1.3.3 Hasil pengujian

Setelah lakukan 10 kali percobaan, maka didapatkan hasil seperti yang ditunjukkan pada Tabel 6.3.

Tabel 6.3 Hasil pembacaan sensor *proximity* kapasitif

Percobaan ke-	Objek	Mengandung Air	Hasil Sensor
1	Tahu goreng	Ya	1020
2	Pensil kayu 2B	Ya	1020
3	Pulpen	Tidak	0
4	Roti tawar	Ya	1019
5	<i>Stapler</i>	Tidak	1021
6	Botol <i>tupperware</i>	Tidak	0
7	Selemba kertas	Ya	0
8	Kantong plastik	Tidak	0

Percobaan ke-	Objek	Mengandung Air	Hasil Sensor
9	Gunting	Tidak	1020
10	Timun	Ya	1020

6.1.3.4 Analisis

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.3, benda akan terdeteksi jika memiliki kapasitansi yang tinggi dari yang dapat dibaca oleh sensor. Jika benda yang dideteksi memiliki kandungan air maka hasil pembacaan dari sensor akan bernilai tinggi mendekati jangkauan maksimal pada sensor. Sampah yang lembap akan terdeteksi dan logam yang tidak mengandung air pun masih dapat terdeteksi oleh sensor ini.

6.2 Pengujian Kinerja Sistem

6.2.1 Tujuan Pengujian

Pengujian ini bertujuan untuk mengetahui tingkat akurasi prediksi yang dilakukan oleh sistem dengan menggunakan algoritme jaringan saraf tiruan *backpropagation*. Data masukan sistem diperoleh dari hasil pembacaan sensor secara langsung.

6.2.2 Prosedur Pengujian

Pengujian akurasi sistem ini dilakukan secara terurut berdasarkan prosedur berikut:

1. Menghubungkan sistem ke komputer melalui kabel USB
2. Melakukan *upload* program ke mikrokontroler Arduino Uno menggunakan PlatformIO IDE
3. Membuka *Serial Monitor* pada PlatformIO IDE
4. Mendekatkan sampah ke seluruh permukaan sensor secara bersamaan
5. Mengamati hasil keputusan yang tampil pada Arduino serta pergerakan servo yang berputar sesuai jenis sampah yang dideteksi.

6.2.3 Hasil Pengujian

Setelah melakukan percobaan sebanyak 10 kali, didapatkan hasil seperti yang ditunjukkan pada Tabel 6.4. Dimana kolom Fitur 1 adalah data hasil pembacaan sensor LDR (tingkat transparansi), fitur 2 adalah data hasil pembacaan sensor *proximity* induktif (kandungan logam), dan fitur 3 adalah hasil pembacaan sensor *proximity* kapasitif (kandungan air).

Tabel 6.4 Hasil uji kesesuaian sistem

Percobaan ke-	Sampah	Fitur 1	Fitur 2	Fitur 3	Kelas Prediksi	Kelas Asli	Kesesuaian
1	Roti Tawar	267	1023	1021	Organik	Organik	Sesuai
2	Tahu Goreng	146	1023	1020	Organik	Organik	Sesuai
3	Mentimun	324	1023	1020	Organik	Organik	Sesuai
4	Kertas HVS	376	1023	27	Organik	Organik	Sesuai
5	Tisu	365	1023	30	Organik	Organik	Sesuai
6	Baterai Kotak	85	2	1018	Anorganik	Anorganik	Sesuai
7	Botol Aqua	832	1023	39	Organik	Anorganik	Tidak Sesuai
8	Gunting kuku	427	15	1020	Anorganik	Anorganik	Sesuai
9	Stapler	189	1	1019	Anorganik	Anorganik	Sesuai
10	Stella	55	2	9	Anorganik	Anorganik	Sesuai

6.2.3.1 Analisis

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.4, dari 10 kali pengujian tersebut dibuat matriks konfusi untuk menghitung kinerja sistem klasifikasi. Tabel 6.5 merupakan matriks konfusi berdasarkan hasil percobaan pada Tabel 6.4.

Tabel 6.5 Analisis hasil pada matriks konfusi

f_{ij}		kelas prediksi (j)	
		Anorganik	Organik
kelas asli (i)	Anorganik	4	1
	Organik	0	5

Berdasarkan matriks konfusi tersebut dapat diketahui akurasi sistem menggunakan Persamaan (2.12) seperti di bawah ini.

$$Akurasi = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

$$Akurasi = \frac{4 + 5}{4 + 1 + 0 + 5} = \frac{9}{10} = 0,9$$

Maka akurasi yang didapatkan berdasarkan perhitungan di atas yaitu 0,9 atau jika dalam persentasenya yaitu 90%.

6.3 Pengujian Performa

6.3.1 Durasi Pelatihan Data

Pada pengujian ini, peneliti akan menguji durasi yang diperlukan pada saat melakukan pelatihan data hingga mencapai tingkat kesalah tertentu. Pelatihan data berupa program python dijalankan pada komputer seperti yang dijelaskan sebelumnya pada analisis kebutuhan. Pelatihan yang digunakan menggunakan 104 data latih, 3 neuron pada lapis masukan, 4 neuron pada lapis tersembunyi dan 2 neuron pada lapis keluaran. Pada pengujian ini peneliti mencoba untuk mengubah-ubah nilai variabel pada tingkat pembelajaran (*learning rate*) dan dijalankan hingga mencapai tingkat kesalah maksimum yang sudah ditentukan. Tingkat pembelajaran yang akan diuji yaitu 0,75, 0,5 dan 0,25. Dengan tingkat kesalahan maksimum yaitu 5, 4,5, 4, 3,5 dan 3.

6.3.1.1 Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui waktu yang diperlukan oleh sistem dalam melakukan pelatihan data menggunakan algoritme *backpropagation*.

6.3.1.2 Prosedur pengujian

Pengujian ini dilakukan secara terurut berdasar prosedur berikut:

6. Menjalankan PyCharm IDE
7. Membuka file *python* pelatihan data
8. Mengubah variabel *learning rate* dan *epoch* sesuai kebutuhan
9. Menyisipkan fungsi untuk mendapatkan waktu sekarang dalam bentuk detik sebagai waktu mulai.
10. Menyisipkan kode untuk menampilkan selisih waktu dari waktu sekarang dengan waktu mulai pada baris setelah pembelajaran dilakukan
11. Menjalankan program
12. Mengamati selisih waktu yang tampil di layar

6.3.1.3 Hasil pengujian

Pengujian durasi dalam pelatihan data ditunjukkan pada Tabel 6.6.

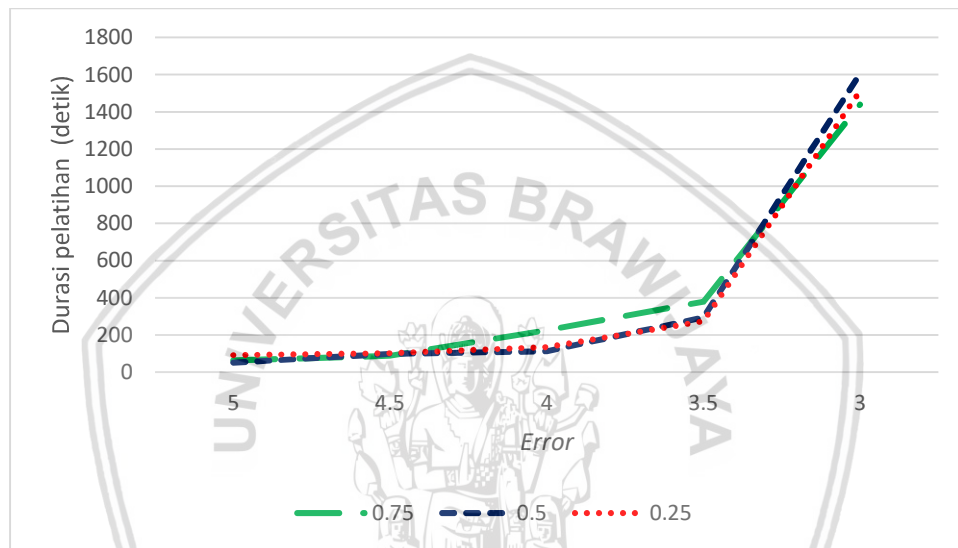
Tabel 6.6 Pengujian durasi pelatihan

<i>Error \ I_rate</i>	0,75	0,5	0,25
5	1m 3,812423s	0m 51,281829s	1m 31,719278s
4,5	1m 27,79647s	1m 39,128284s	1m 41,256878s
4	3m 46,518105s	1m 52,714855s	2m 14,805599s
3,5	6m 19,74596s	4m 55,188083s	4m 34,218623s

Error \ l_rate	0,75	0,5	0,25
3	23m 58,984101s	26m 44,593236s	25m 12,015369s

6.3.1.4 Analisis

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.6, dapat disimpulkan bahwa durasi sistem dalam melakukan pelatihan tidak begitu dipengaruhi oleh nilai *learning rate* namun lebih tingkat kesalahan yang diraih. Semakin kecil tingkat kesalahan yang dicapai durasi waktu yang diperlukan juga akan semakin lama. Gambar 6.1 menunjukkan bagaimana perubahan data dalam bentuk diagram garis.



Gambar 6.1 Diagram analisis pengujian durasi selama pelatihan data

6.3.2 Durasi Prediksi

6.3.2.1 Tujuan pengujian

Pengujian ini bertujuan untuk mengetahui durasi yang diperlukan oleh mikrokontroler pada sistem untuk memprediksi satu jenis sampah menggunakan jaringan saraf tiruan.

6.3.2.2 Prosedur pengujian

Pengujian ini dilakukan secara terurut berdasarkan prosedur berikut:

1. Membuka PlatformIO IDE
2. Menyisipkan baris kode pada file program Arduino untuk menyimpan waktu sekarang dalam bentuk mili sekon sebelum klasifikasi dilakukan
3. Menyisipkan baris kode pada file program Arduino untuk menghitung dan menampilkan selisih waktu sekarang dengan waktu mulai
4. Menghubungkan mikrokontroler Arduino sistem ke komputer melalui kabel USB

5. Melakukan *upload* program menggunakan PlatformIO IDE ke mikrokontroler Arduino menggunakan PlatformIO IDE
6. Membuka *Serial Monitor* pada PlatformIO IDE
7. Mendekatkan sampah ke seluruh permukaan sensor secara bersamaan
8. Melihat selisih waktu berdasarkan nilai yang tampil di *serial monitor*

6.3.2.3 Hasil pengujian

Berdasarkan pengujian yang telah dilakukan dengan 10 kali percobaan, maka didapatkan hasil yang ditunjukkan pada Tabel 6.6.

Tabel 6.7 Hasil pengujian durasi prediksi

Percobaan ke-	Objek	Kelas Prediksi	Durasi (ms)
1	Pulpen	Anorganik	43
2	Pensil kayu 2B	Organik	43
3	Kertas	Organik	42
4	Roti tawar	Organik	43
5	Stapler	Anorganik	43
6	Botol <i>tupperware</i>	Anorganik	43
7	Spidol	Anorganik	43
8	Kantong plastik	Anorganik	43
9	Gunting	Anorganik	43
10	Timun	Organik	43

6.3.2.4 Analisis

Berdasarkan hasil pengujian yang ditampilkan pada Tabel 6.6, dapat diperoleh hasil rata-rata durasi menggunakan Persamaan (2.13). Berikut merupakan hasil perhitungannya:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n X_i$$

$$\bar{x} = \frac{1}{10} (43 + 43 + 42 + 43 + 43 + 43 + 43 + 43 + 43 + 43)$$

$$\bar{x} = \frac{429}{10} = 42,9$$

Dimana \bar{x} adalah rata-rata durasi, X_i adalah durasi prediksi ke-i (dalam milisekon) dan n adalah jumlah seluruh data yaitu 10 data. Berdasarkan hasil perhitungan tersebut, dapat disimpulkan bahwa sistem dapat melakukan prediksi dengan sangat cepat karena hanya membutuhkan waktu 42,9ms.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil dari bab perancangan, implementasi, pengujian dan analisis yang telah dilakukan pada pengujian ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Pada penelitian ini telah dibuat sistem dengan klasifikasi sampah organik dan anorganik dengan menggunakan metode jaringan saraf tiruan. Sistem dapat membedakan jenis sampah berdasarkan data sensor LDR, *proximity* induktif, serta *proximity* kapasitif menggunakan algoritme jaringan saraf tiruan *backpropagation*.
2. Tingkat akurasi yang berdasarkan pengujian pada bab sebelumnya, sistem dapat melakukan klasifikasi dengan menggunakan metode jaringan saraf tiruan dengan tingkat keakuratan mencapai 90%.
3. Berdasarkan pengujian durasi waktu yang dilakukan pada bab sebelumnya, dari 10 kali percobaan dengan beberapa sampah dan jenis yang berbeda-beda, dalam perhitungan prediksi sistem hanya perlu menempuh waktu rata-rata sekitar 42,9ms.

7.2 Saran

Berdasarkan percobaan yang telah dilakukan, penelitian ini masih memiliki banyak kekurangan. Untuk penelitian selanjutnya, peneliti perlu memberikan saran sebagai berikut:

1. Jumlah data latih, jumlah lapisan dan neuron pada lapisan tersembunyi dapat diperbanyak hingga dapat meningkatkan akurasi sistem.
2. Penelitian selanjutnya dapat menggunakan fungsi aktivasi yang berbeda sebagai pembanding tingkat akurasi yang dihasilkan oleh sistem.
3. Penelitian selanjutnya perlu mencoba beberapa kombinasi arsitektur seperti banyaknya lapis tersembunyi yang digunakan dan variasi jumlah unit dalam setiap lapis tersembunyi.

DAFTAR PUSTAKA

- Abdi, H. (2007). *Encyclopedia of Measurement and Statistics*. CA, USA: Thousand Oaks.
- Agarwal, T. (2015). *What is a Light Dependent Resistor and Its Applications*. Diambil kembali dari EdgeFX kits: <http://www.edgefxkits.com/blog/light-dependent-resistor-ldr-with-applications/>
- Badan Pusat Statistik. (2017, April 5). *Proyeksi Penduduk Indonesia Berdasarkan Hasil Sensus Penduduk 2010*. Diambil kembali dari <https://bps.go.id/>
- Dharmawan, H. A. (2017). *Mikrokontroller: Konsep Dasar dan Praktis*. Malang: Universitas Brawijaya Press.
- Djohan, A., & Halim, D. (2013). *Pengelolaan Limbah Rumah Sakit*. Jakarta: Salemba Medika.
- Hui, L. K. (2012). *Basic – Pulse Width Modulation (PWM)*. Diambil kembali dari Tutorials by Cytron: <https://tutorial.cytron.io/2012/01/14/basic-pulse-width-modulation-pwm/>
- Kementerian Lingkungan Hidup dan Kehutanan. (2015). *Rangkaian HLH 2015 – Dialog Penanganan Sampah Plastik*. Diambil kembali dari Kementrian Lingkungan Hidup: <http://www.menlh.go.id/rangkaian-hlh-2015-dialog-penanganan-sampah-plastik/>
- Kretschmar, M. (2009). *Capacitive Sensor Operation Part 1: The Basics*. Diambil kembali dari Sensors Magazine: <https://www.sensormag.com/components/capacitive-sensor-operation-part-1-basics>
- Kristanto, A. (2004). *Jaringan Saraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasi)*. Yogyakarta: Gava Media.
- Menke, H. (2014). *Basic Operating Principle of an Inductive Proximity Sensor*. Diambil kembali dari Automation Insights: <https://automation-insights.blog/2014/03/05/basic-operating-principle-of-an-inductive-proximity-sensor/>
- Mulyanto, E., Suhartono, V., & Sutojo, T. (2011). *Kecerdasan Buatan*. Yogyakarta: ANDI.
- Munir, R. (2004). *Pengolahan Citra Digital*. Bandung: Informatika.
- Nuraini, N. U. (2017). Perancangan Dan Implementasi Sistem Klasifikasi Jenis Sampah Rumah Tangga Dengan Menggunakan Metode Naïve Bayes. *Repositori Jurnal Mahasiswa PTIIK UB Vol. 9 No. 3*.
- Oladapo, B. I. (2016). Model design and simulation of automatic sorting machine using proximity sensor. *Science Direct*, 1452-1456.

- Prasetya, R. (2017). Implementasi Jaringan Syaraf Tiruan Backpropagation Untuk Memprediksi Ketinggian Air (Studi Kasus: Sungai Ciliwung). *Jurnal String* Vol. 1 No.3.
- Prasetyo, E. (2012). *Data mining konsep dan aplikasi menggunakan MATLAB*. Yogyakarta: Andi Offset.
- Puspitaningrum, D. (2006). *Pengantar Jaringan Syaraf Tiruan*. Yogyakarta: Andi.
- Supranto, J. (2008). *Statistik Teori dan Aplikasi*. Jakarta: Erlangga.
- Susanto, A. (2003). *Pengenalan Komputer*. Dipetik Januari 2, 2018, dari http://nyoman.staf.narotama.ac.id/files/2012/01/arief_pengenalankomputer.pdf
- Yohandri, & Asrizal. (2016). *Elektronika Dasar*. Jakarta: Prenada Media.

